

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO
FAKULTETA ZA MATEMATIKO IN FIZIKO

Luka Androjna

Optimizacija konverzij v programatičnem oglaševanju

DIPLOMSKO DELO

INTERDISCIPLINARNI UNIVERZITETNI
ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN MATEMATIKA

MENTOR: doc. dr. Lovro Šubelj

SOMENTOR: Andraž Tori

SOMENTOR: Davorin Kopic

Ljubljana, 2019

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge: Optimizacija konverzij v programatičnem oglaševanju

V diplomski nalogi preučite domeno programatičnega spletnega oglaševanja v realnem času. Posebej se osredotočite na optimizacijo nakupovanja spletnega oglasnega prostora in konverzij, ki se pri tem dogajajo. Primerjajte različne enostavne pristope za optimizacijo konverzij in predlagajte lastno rešitev, ki se najbolj izkaže v praksi. Rešitev implementirajte in preizkusite v produkcijskem okolju podjetja Zemanta d.o.o. tekom daljšega časovnega obdobja. Z uporabo statističnega preverjanja hipotez preverite ali je razvita rešitev značilno bolj uspešna od obstoječega sistema.

Zahvaljujem se mentorju, doc. dr. Lovru Šublju, za usmerjanje in napotke za izdelavo diplomske naloge, somenotrijema Andražu Toriju in Davorinu Kopicu, ki sta mi odprla vrata v industrijo in omogočila delo na področju podatkovne znanosti. Zahvaljujem se tudi staršem, Nini in vse drugim, ki so mi pomagali na moji akademski poti.

”As data scientists, our job is to extract
signal from noise.”

- Daniel Tunkelang

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Programatično oglaševanje in konverzije	5
2.1	Programatično oglaševanje	6
2.2	Udeleženci v RTB	10
2.3	Konverzije	11
2.4	Problem optimizacije	13
3	Orodja, metode in podatki	15
3.1	Orodja	15
3.1.1	Google BigQuery	15
3.1.2	Python 3	16
3.2	Metode	19
3.2.1	Testiranje	19
3.2.2	Iskanje in klasificiranje sledilnih pikslov	20
3.2.3	Analiza podatkov	21
3.3	Podatki	22
4	Opis dela	23
4.1	Seznam dovoljenih XPT	23
4.1.1	Iskanje najustreznejše razvrstitve	24

4.1.2	Določanje spodnje meje	35
4.1.3	A/B testiranje v produkciji	38
4.2	Seznam prepovedanih uporabniških agentov	38
5	Rezultati in diskusija	41
6	Zaključek in prihodnje delo	45
	Literatura	49

Seznam uporabljenih kratic

kratica	angleško	slovensko
CPA	cost per acquisition	cena za akvizicijo
CPC	cost per click	cena na klik
CPM	cost per mille	cena na tisoč prikazov
CVR	conversion rate	razmerje med številom klikov in konverzij
DSP	demand-side platform	platforma na strani povpraševanja
IT	information technology	informacijska tehnologija
KPI	key performance indicator	ključni kazalnik uspešnosti
MVP	minimum viable product	najosnovnejši sprejemljivi produkt
RTB	real-time bidding	draženje v realnem času
SSP	supply-side platform	platforma na strani ponudbe
XPT	exchange, publisher, tagid combination	kombinacija oglasne borze, spletne strani in identifikatorja oglasnega mesta

Povzetek

Naslov: Optimizacija konverzij v programatičnem oglaševanju

Avtor: Luka Androjna

Založniki na spletu na svojih spletnih straneh služijo z oglaševanjem. Oglesi so namenjeni bralcem vsebin založnikovih strani, za njih pa plačujejo oglaševalci. Ti ponavadi prikazujejo oglase, da bi izboljšali ugled znamke ali pa pridobili uporabnike, ki opravijo določene akcije. Oglaševalci si želijo, da preko klika na oglas k njim pride obiskovalec, ki jim bo nekaj doprinesel, na primer ustvaril nov uporabniški račun, se prijavil na njihovo glasilo, opravil nakup itd. Takim akcijam pravimo konverzije. Cilj programatičnega oglaševanja je optimizirati nakupovanje oglasnega prostora tako, da se zgodijo kliki na oglas in konverzije. V tej nalogi si bomo ogledali različne pristope, kako znižati ceno za akvizicijo in povišati razmerje med kliki in konverzijami. Problema se lotimo z uporabo seznama dovoljenih kombinacij oglasne borze, spletne strani in identifikatorja oglasnega mesta ter seznamom prepovedanih uporabniških agentov. Z A/B testom, ki je potekal v produkcijskem okolju in vplival na kupovanje v realnem času podjetja Zemanta d.o.o., smo potrdili, da s kupovanjem prometa na boljših oglasnih mestih pridobimo boljše razmerje med kliki in konverzijami ter nižjo ceno za posamezno konverzijo. Opazimo tudi, da je nižja tudi cena klika, čeprav je cena prikazov višja.

Ključne besede: oglaševanje, podatkovna znanost, podatkovna analiza.

Abstract

Title: Optimization of Conversions in Programmatic Advertising

Author: Luka Androjna

Publishers on the web monetize their businesses by running advertisement on their websites. Ads are targeted at their readers and paid for by advertisers. Advertisers generally show ads to either improve their brand perception or to get users to make some specific action. They want to get engaging users onto their product's landing page and take actions that generate value for the advertiser, for example registering a new account, subscribing to a newsletter, making a purchase etc. Those actions are called conversions. Optimizing buying of advertising to induce clicks and conversions is the business of programmatic advertising. We will look at different approaches for lowering the CPAs and raising conversion rates. We approach the problem with the use of a whitelist containing combinations of exchanges, publisher websites and tag ids, and a blacklist of useragents. We conducted an A/B test, which was running in an online production environment and affected bidding in real time for Zemanta d.o.o., that showed buying traffic on premium ad space gives us better conversion rates and lower prices per acquisition. We also noticed that prices per click dropped even though the average price per impression went up.

Keywords: advertising, data science, data analysis.

Poglavje 1

Uvod

Oglas je neosebno tržno sporočilo, ki je namenjeno promociji oziroma prodaji izdelka, storitve ali ideje. Oglaševalci so običajno podjetja, ki želijo promovirati svoje izdelke ali storitve. Oglaševanje ni namenjeno določenemu posamezniku, temveč množici, ki spada v specifičen segment trga. Distribucija oglasov poteka preko različnih množičnih medijev. Tradicionalni mediji so časopisi, revije, televizija, radio, oglasi na prostem in po pošti. Nove medije pa predstavljajo blogi, brskalniki, socialna omrežja in spletne strani.

Prodajanje prostora za oglaševanje je eden od glavnih načinov, kako založniki monetizirajo svoje spletne portale. Prikazovanje oglasov na spletnih straneh je dvorezen meč, saj po eni strani nudi založniki vir zaslužka in omogoča uporabnikom brezplačen dostop do vsebin, po drugi strani pa lahko s prekomernim oglaševanjem poslabšajo uporabniško izkušnjo. Spletno oglaševanje se je začelo kmalu po začetku interneta z oglasi v spletni pošti. Proti koncu leta 1994 se je pojavila prva oglasna pasica (angl. *banner ad*), njen oglaševalec je bil *AT&T*, objavljena je bila na strani digitalne publikacije *HotWired* [18]. Publikacija je na svoji strani implemetirala in rezervirala prostore za oglase na podoben način, kot so jih imele tiskane revije in časopisi. Tem prostorom so rekli *banner ad*. Cena oglaševanja na spletnih straneh je bila stvar dogovora med oglaševalcem in založnikom, cena je bila določena vnaprej in je bila vezana na časovno obdobje, v katerem je bil oglas prikazan

na strani.

Do danes se je plačniški model spremenil, uporabljata se modela plačaj-za-klik (angl. *pay-per-click*) in plačaj-za-prikaz (angl. *pay-per-impression*). Ko uporabnik odpre spletno stran, se v času, ko se stran nalaga, sproži dražba za oglaševalske prostore na strani. Torej dražba poteka v realnem času, kar je boljše tako za založnike in oglaševalce kot tudi za obiskovalce strani. Oglaševalci lahko s pridobljenimi podatki o spletni strani in njenem obiskovalcu podajajo boljše ponudbe nakupnih cen, saj lažje ocenijo, koliko jim je vreden prikaz oglasa na določeni strani za specifičnega obiskovalca, obiskovalcu pa se lahko izboljša uporabniška izkušnja vsaj do te mere, da se mu prikaže oglas iz področja, ki ga zanima.

Zaradi zmožnosti ciljanja na določene segmente prometa na internetu, se čedalje več podjetji odloča za spletni oglaševanje. Posledično je spletno oglaševanje ena od najhitreje rastočih panog v IT industriji [20]. V mobilnem in prikaznem oglaševanju je bil največji del razvoja usmerjen v razvijanje draženja v realnem času (RTB).

Cilj spletnih oglaševalcev je preko objavljenih oglasov na svoje ciljne strani pridobiti uporabnike, ki opravijo doprinosno dejavnost. Med te sodi registracija novega računa, nakup izdelka ali storitve itd. Te dejavnosti imenujemo konverzije. V tej diplomski nalogi se bomo posvetili optimizaciji kupovanja oglasnega prostora tako, da bo verjetnost konverzije čim višja.

Problem poskušamo rešiti z uporabo seznama dovoljenih kombinacij oglasne borze, spletne strani in identifikatorja oglasnega mesta ter seznama prepovedanih uporabniških agentov. Pri implemetaciji seznamov bomo uporabljali znanja iz področja podatkovne znanosti, kot so: poznavanje in uporaba velikih podatkovnih baz, poznavanje programskega jezika *Python* in njegovih knjižnic, statistično analizo in obdelavo podatkov v *Python*-u, prevod analize iz *Python*-a v *SQL*, da lahko obdelamo večje količine podatkov, implementiranje A/B testov in domenskega znanja o programatičnem oglaševanju. A/B test bo potekal na podjetju Zemanta d.o.o. v produkcijskem okolju v realnem času in bo vplival na kupovanje prometa.

V nadaljevanju naloge se najprej posvetimo razlagi in opisu industrije, da predstavimo ozadje programatičnega oglaševanja. Dotaknemo se samega programatičnega oglaševanja in udeležencev v dražbah v realnem času. Poglavje pa zaključimo z opisom konverzij in problema, ki ga poskušamo rešiti. Temu sledi opis orodja, metod in podatkov, katere uporabljamo v raziskovalnem delu. Nato se osredotočimo na opis postopka dela, kjer predstavimo možno rešitev in način izpeljave. Opisu dela sledi interpretacija rezultatov.

Poglavje 2

Programatično oglaševanje in konverzije

Spletno oglaševanje je oblika oglaševanja, ki uporablja splet kot medij za prikazovanje promocijskih sporočil potrošnikom. Spletno oglaševanje zajema marketing preko spletne pošte, marketing z iskalnikom, marketing preko družbenih medijev, več različnih tipov prikaznih oglasov in mobilno oglaševanje. Podobno kot pri drugih oblikah oglaševanja zajema spletno oglaševanje založnika, ki umesti oglas v vsebino svoje spletne strani, in oglaševalca, ki priskrbi oglas za objavo. Zajema pa lahko tudi oglaševalske agencije, ki pomagajo pri ustvarjanju in umestitvi oglasov, oglasni strežnik, ki je odgovoren za dostavljanje oglasov in sledi statistiki, in oglaševalske podružnice, ki opravljajo promocijska dela za oglaševalca. Spletno oglaševanje se deli na oglaševanje blagovnih znamk (angl. *brand advertising*) in oglaševanje z neposrednim odzivom (angl. *direct response advertising*). Fokus oglaševanja blagovnih znamk je graditi prepoznavnost in ugled znamke s simboli in slogani. Na drugi strani pa je oglaševanje z neposrednim odzivom, ki je bolj osredotočeno na optimiziranje cene na klik in posledično konverzij. V tej analogi se osredotočimo na slednje.

2.1 Programatično oglaševanje

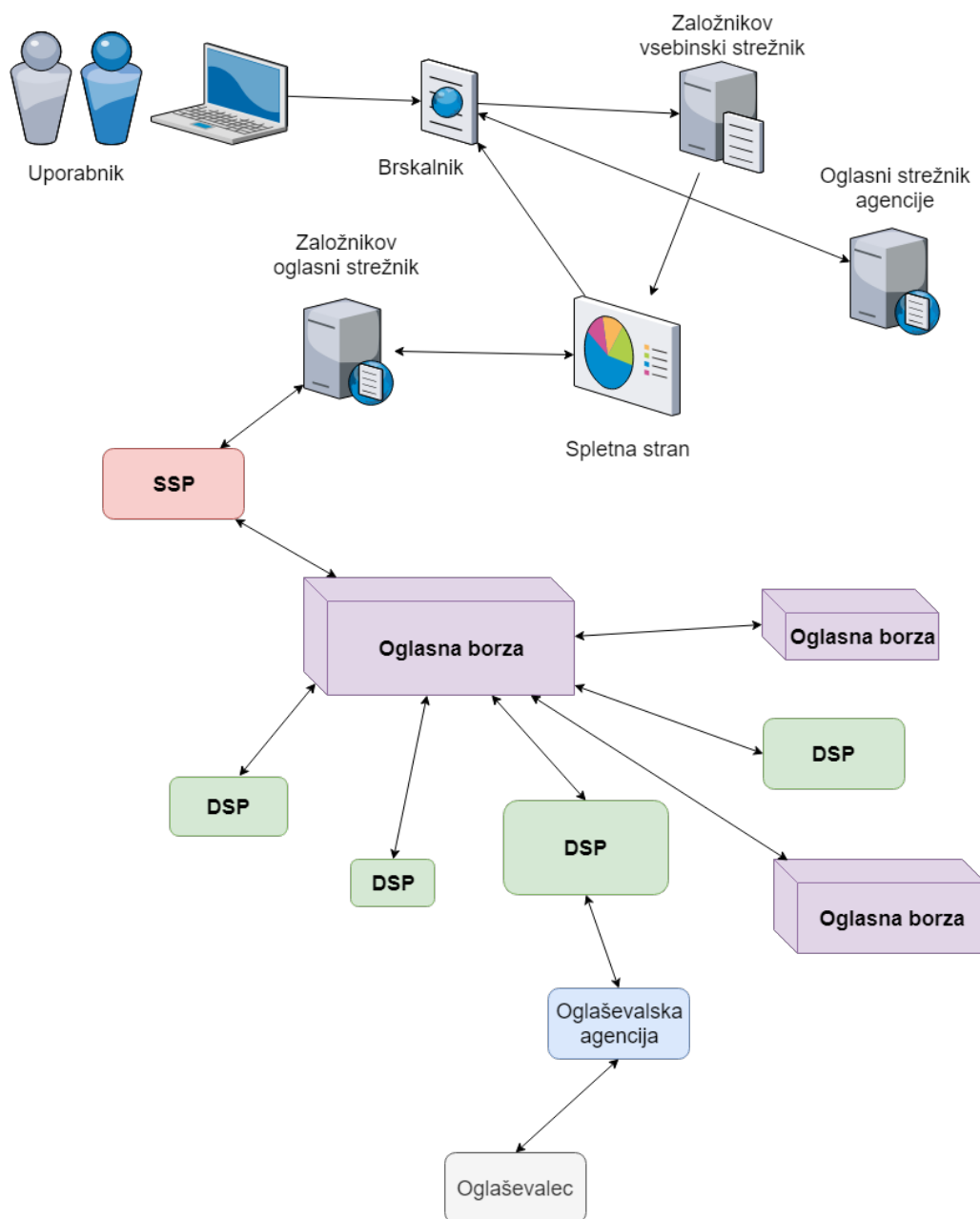
Programatično oglaševanje je en od pristopov spletnega oglaševanja, pri kateri so umestitve na posamezne prikaze oglasov kupljene v realnem času s pomočjo programske opreme. Za programatično oglaševanje veljata načeli granularnosti in avtomatizacije. Granularnost sledi iz sposobnosti obravnave na nivoju posameznega prikaza oglasa, na parametrih oglasa in okolja, v katerega naj bi bil umeščen. Granularnost omogoča oglaševalcem nov način optimizacije proračuna, saj se lahko odločijo za nakup ob posamezni priložnosti za umestitev prikaza oglasa na podlagi ocene, ki jo pripišejo prikazu. Avtomatizacija pa omogoča ocenjevanje zaloge oglasnega prostora in njegove vrednosti za oglaševalca.

Draženje v realnem času je sistem posredovanja medijev, ki omogoča, da se posamezen prikaz oglasa proda na dražbi v realnem času. To omogočajo programatične dražbe, ki so podobne tistim iz finančnega trga. RTB omogoča naslovno oglaševanje (angl. *addressable advertising*), sposobnost neposrednega serviranja oglasov potrošnikom glede na njihove demografske, psihološke in vedenjske lastnosti [1].

Ko obiskovalec odpre spletno stran, se za oglasne prostore, ki se nahajajo na tej strani, priredi dražba. Dražbe prirejajo platforme na strani ponudbe (SSP) tako, da obvestijo platforme na strani povpraševanja (DSP), da so na voljo oglasna mesta na dani spletni strani, poleg tega pa jim pošljejo še podatke o strani in uporabniku (v standardnem OpenRTB formatu). DSP-ji iz dobljenih parametrov ocenijo, kateri oglas iz svojega sistema je najbolj primeren za prikaz in koliko so pripravljeni plačati za prikaz izbranega oglasa. DSP-ji na oglasno borzo [14] (angl. *ad exchange*) pošljejo oglas, ki ga želijo prikazati, z njegovimi meta podatki (slika, naslov, kategorija itd.) in ponudbo o nakupni ceni, ki so jo pripravljeni plačati. Tako DSP-ji umeščajo svoj inventar oglasov.

Dražba za oglasni prostor naj bi se zaključila, preden se naloži stran, ki jo je uporabnik obiskal, zato imajo DSP-ji časovno omejitev za odgovor na SSP-jevo zahtevo za draženje (angl. *bid request*). Ta omejitev znaša med 10

in 100 milisekund [12], kar je odvisno od borze. Ta čas zajema tudi latenco omrežja. Zaradi časovne omejitve vsak udeleženec v dražbi le enkrat poda ponudbo za svoj oglas.



Slika 2.1: Diagram RTB sistema.

Ko uporabnik obiše spletno stran, se odpre povezava na založnikov vsebinski strežnik, ki najde in servira stran z vsebino, ki jo je obiskovalec želel naložiti. Ta stran vsebuje povezavo na založnikov oglasni strežnik. Ta se poveže na SSP, ki oplemeniti zahtevo za dražbo s podatki o uporabniku, in jo pošlje naprej na borzo z oglasi. Ta je povezana s sistemi za nakup oglasnih prikazov, to so DSP-ji, oglaševalska omrežja in celo druge oglaševalske borze. V tej točki se začne dražba med partnerji prvotne borze z oglasi. Udeleženec dražbe, ki zmagaja, borzi pošlje navodila za serviranje oglasa. Borza zahtevo prenese nazaj na SSP, ta pa zahtevo prenese na oglasni strežnik. Oglasni strežnik obvesti brskalnik, od kod naložiti oglas. Ta je ponavadi na oglasnem strežniku agencije, ki z zahtevo za nalaganje oglasa beleži prikaze. Povezave so navedene v diagramu na sliki 2.1 [22].

Dražbe so zasnovane po modelu Vickreyjevih dražb [17] (angl. *Vickrey auction*). Dražitelji predložijo ceno, za katero mislijo, da je najbolj poštena, nevedoč, koliko bodo ponudili ostali. Zmagovalec Vickreyjeve dražbe plača ceno druge najvišje ponudbe namesto svoje. Tej implemetaciji rečejo tudi dražba druge cene (angl. *second-price auction*). V primeru, ko zmagovalec plača ceno svoje ponudbe, govorimo o dražbah prve cene (angl. *first-price auction*). Prevladujoč model v RTB je model dražb druge cene [18], ker spodbuja pravično licitiranje, saj dražitelji ponudijo najvišjo ceno, ki so jo pripravljeni plačati. Zavedajo se, da je cena prikaza v primeru zmage nižja od tiste, ki so jo predlagali.

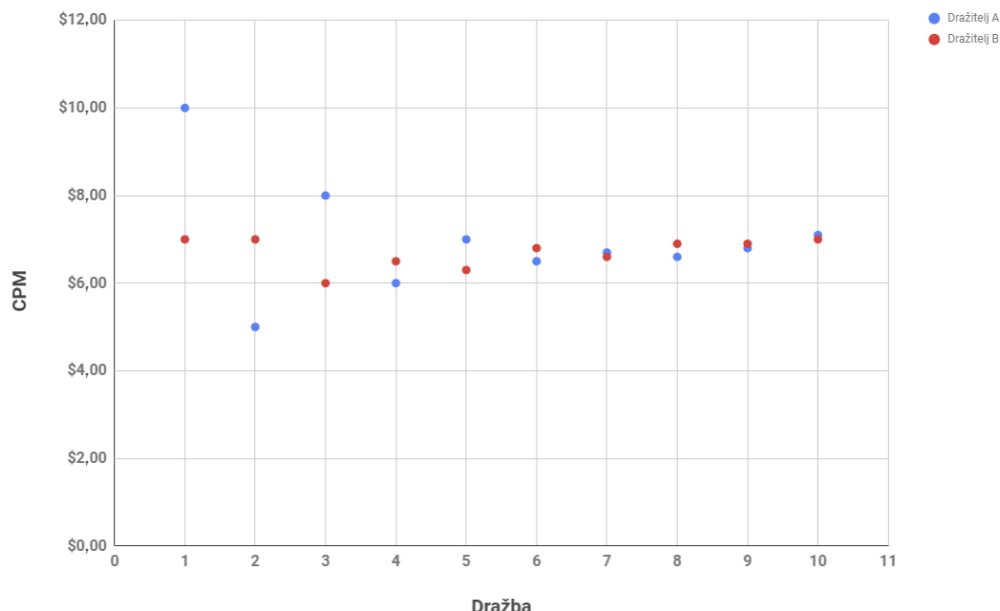
Intuicijo, zakaj večina oglasnih borz uporablja za model draženja dražbe druge cene, lahko razložimo z naslednjim poenostavljenim primerom. Na dražbi prve cene sodelujeta le dva dražitelja, za katera privzamemo, da dinamično prilagajata strategijo ponudb in imata različni najvišji ponudbi za prikaz oglasov, in predpostavimo, da sodelujeta na zaporednih dražbah za isti oglasni prostor, bo po nekaj dražbah začel zmagovati dražitelj z višjo najvišjo ponudbo. Cena, ki jo bo moral plačati, pa konvergira k zgornji meji najvišje ponudbe drugega dražitelja in najmanjši denarni enoti, ki jo sistem prizna.

Denimo, da imamo dražitelja A, ki je za prikaz oglasa pripravljen plačati \$10 in dražitelja B, ki je za prikaz svojega oglasa pripravljen plačati \$7 ter določen prostor, za katerega se izvajajo zaporedne dražbe. Na prvi dražbi dražitelja nimata podatka o povprečni ceni prostora, zato ponudita najvišjo ceno znotraj svojega proračuna. Dražbo zmaga dražitelj A, s tem pridobi nekaj znanja o oglasnem prostoru, povprečna cena prikaza oglasa je pod \$10. Dražitelj B ne prejme informacij, ki bi mu pomagale do bolj optimalnega ponujanja. Ker je dražitelj A zmagal, na naslednji dražbi poskusi s ponudbo nižje nakupne cene, na primer ponudi \$5. Dražitelj B ponovno ponudi \$7 in zmaga. Dražitelj A tako izve, da je optimalna cena, ki jo ponudi za oglasni prostor, med \$5 in \$10. Dražitelj B bo na naslednji dražbi poskusil z nižjo ponudbo. Izmenjevanje zmag, spuščanje in dvigovanje cene se nadaljuje, dokler dražitelj A ne ugotovi, da mora ponuditi več kot \$7 in manj kot \$10. Od te točke naprej stavi \$7,01 in zmaga vsako dražbo. Torej dražbe prve cene konvergirajo proti dražbam druge cene (glej tabelo 2.1 in graf na sliki 2.2).

Dražba	Dražitelj A	Dražitelj B
1	\$10,00	\$7,00
2	\$5,00	\$7,00
3	\$8,00	\$6,00
4	\$6,00	\$6,50
5	\$7,00	\$6,30
6	\$6,50	\$6,80
7	\$6,70	\$6,60
8	\$6,60	\$6,90
9	\$6,80	\$6,90
10	\$7,01	\$7,00

Tabela 2.1: Prikaz nihanja cen (prikazanih s CPM skozi dražbe).

Nihanje licitiranih CPM skozi dražbe



Slika 2.2: Prikaz nihanja cen (prikazanih s CPM skozi dražbe) v obliki grafa.

2.2 Udeleženci v RTB

Platforme na strani ponudbe (angl. *supply-side platform*) so programska oprema, ki omogoča spletnim založnikom upravljati s svojim inventarjem oglasnega prostora, polnjenje prostora z oglasi in prejemanja plačil za prikaz oglasov. Večina založnikov uporablja SSP za avtomatizacijo in optimizacijo prodaje oglasnega prostora. SSP-ji pošiljajo informacije o potencialnih prikazih na borzo z oglasi, kjer so licitirani DSP-jem. S tem omogočajo maksimiziranje donosa založnikovega oglasnega prostora.

Platforme na strani povpraševanja (angl. *demand-side platform*) so programska oprema, ki omogoča oglaševalcem upravljanje računov na podatkovnih in oglaševalskih borzah skozi en vmesnik. DSP-ji v RTB-ju služijo kot pooblaščen udeleženec dražbe za oglaševalca. DSP-ji omogočajo svojim strankam ustvarjanje ciljnih občinstev, za katere lahko nastavljajo ponudbe za oglaševalska mesta. DSP-ji ponujajo optimizirano porabo denarja za oglaševanje glede na izbran ključni kazalnik upešnosti (KPI), ki ga iz-

bere stranka. Med KPI-je sodijo učinkovita cena na klik, učinkovita cena na akvizicijo, stopnja obiskov strani in podobno.

DSP-ji ponujajo veliko funkcij, ki so jih klasično omogočala oglaševalska omrežja. Med njih sodi dostop do širokega inventarja oglasnih mest, sposobnost upodabljanja in prikaza oglasa, sodelovanje na dražbah v realnem času, sledenje oglasom in optimizacijo kupovanja glede na preteklo stanje. S tem DSP-ji omogočajo oglaševalcem, da maksimizirajo vpliv svojih oglasov.

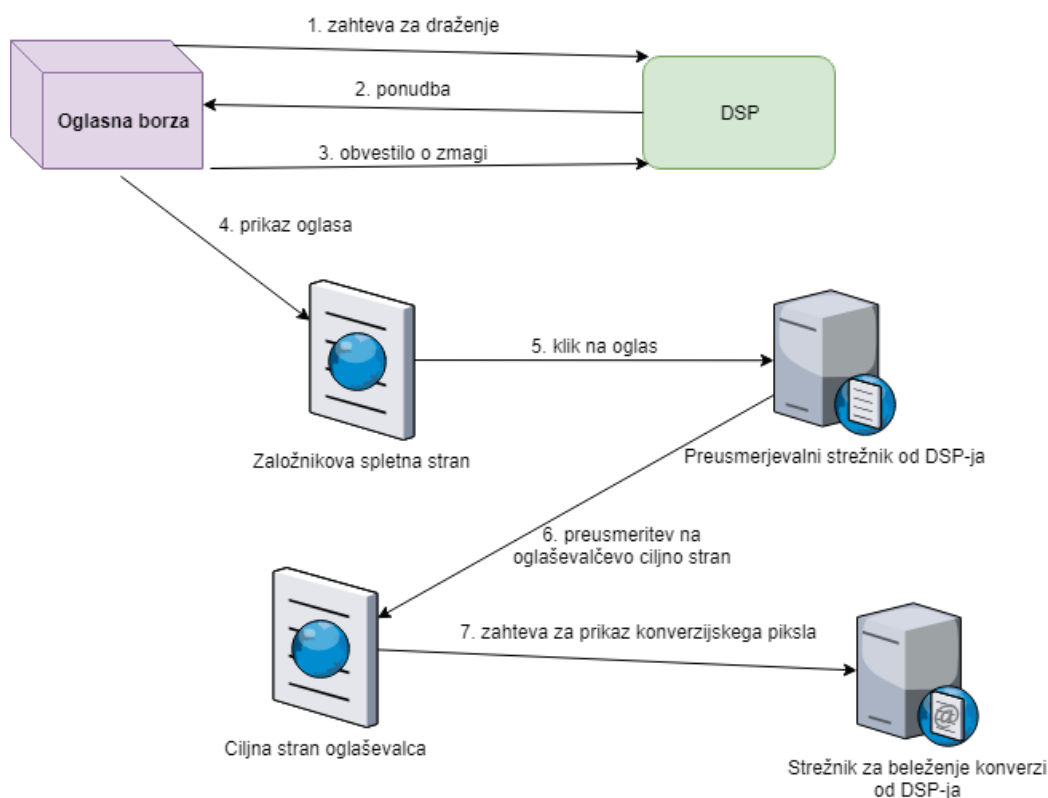
2.3 Konverzije

Programatično kupovanje oglasnega prostora je na spletu osredotočeno na učinkovitost. Za založnike vrednost temelji na prodaji oglasnega prostora, vrednost oglaševanja na spletu pa se za oglaševalca pokaže šele po kliku uporabnika na prikazan oglas. Cilj oglaševalcev je pridobiti nove stranke ali uporabnike, ki na oglaševalčevi spletni strani opravi donosno dejavnost. Takim dejavnostim pravimo konverzije.

Konverzije delimo na lahke, srednje in težke. Med lahke sodijo obisk ciljne strani, stran odprta za določeno časovno obdobje, ogledan posnetek in podobne dejavnosti, kjer uporabnik nima dodatnega dela, kot je na primer vpisovanje osebnih podatkov. Med srednje konverzije štejemo akcije, ki od uporabnika zahtevajo, da vloži nekaj truda, na primer vnos elektronskega naslova, izpolnjen obrazec, registracija novega uporabniškega računa in podobno. Med težke konverzije sodijo registracije z dvofaktorsko avtentikacijo, izpolnjevanje polj za dokazovanje, da nismo robot (angl. *captcha*), nakupi izdelkov in naročila storitev.

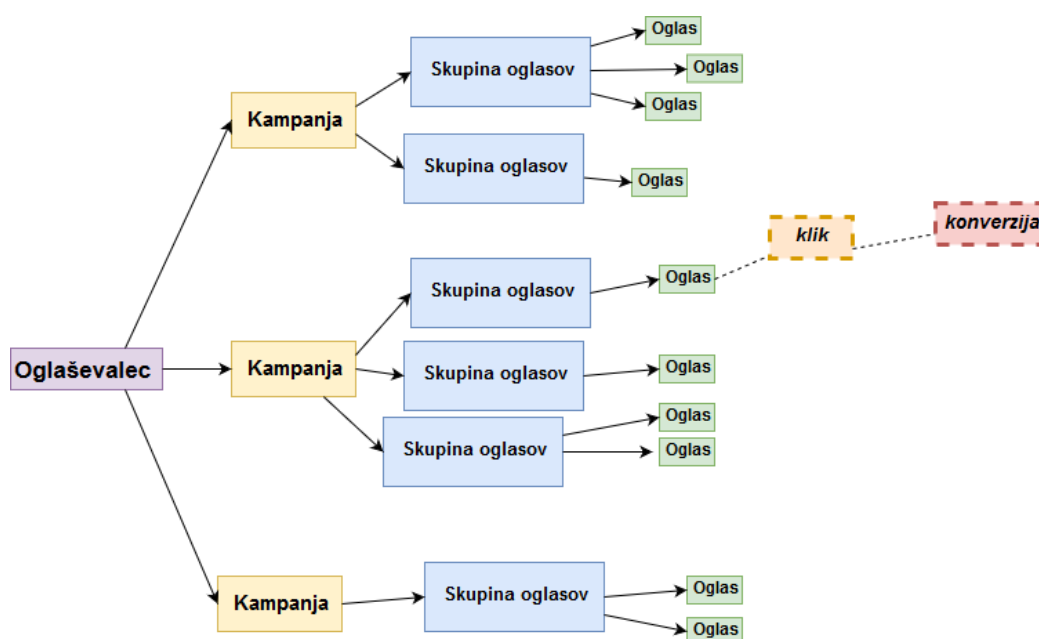
Da si DSP lahko pripiše konverzijo, mora slediti akcijam, ki sledijo prikazu oglasa. To počnejo s preusmerjanjem prek strežnikov in slikovnimi točkami oziroma piksli (angl. *pixel*). Po oddani ponudbi za nakup oglasnega mesta dobi DSP, ki je dražbo zmagal, obvestilo o zmagi, s katerim si zabeleži prikaz. V primeru, ko prikaz privede do klika na oglas, je uporabnik preusmerjen na DSP-jev preusmerjevalni strežnik, ki zabeleži klik in preusmeri

uporabnika na ciljno stran oglaševalca. Ciljne strani imajo vgrajeno kodo ali pa *HTML* slikovni objekt, ki pošlje zahtevo na DSP-jev strežnik in tako zabeleži obisk ciljne strani (lahka konverzija). Ostali tipi konverzij se beležijo na podoben način, vsaka akcija ima svoj piksel, ki se sproži ob zaključeni akciji, na primer, ko uporabnik zaključi nakup, je preusmerjen na stran za pregled naročila ali pa na stran z zahvalo za nakup, te pa vsebujejo piksele, ki DSP-ju povedo, da se je zgodila konverzija. Za pošteno pripisovanje konverzij mora DSP slediti tudi uporabniku od prikaza do konverzije, da lahko zagotovi, da je isti uporabnik videl oglas, nanj kliknil, obiskal oglaševalčevo cilno stran, in opravil konverzijo. Sledenje uporabniku poteka preko uporabniških piškotkov [21], ki vsebujejo unikatno identifikacijsko številko uporabnika. Za vsako identifikacijsko številko iz piškotkov, ki jih DSP vidi na svojem strežniku, naredi preslikavo v uporabniško številko v svojem sistemu, preko katere lahko spremlja uporabnika, dokler ta ne pobriše svojih piškotkov.



Slika 2.3: Diagram sledenja oglasu.

Kot v navadnem oglaševanju imajo tudi v programatičnem oglaševanju oglaševalci kampanje, ki vsebujejo skupine oglasov. V vsaki skupini oglasov je nabor oglasov, za katere kupujejo oglaševalski prostor (prikazano v sliki 2.4). Posamezni kliki in konverzije pripadajo kampanjam, ker so vezani na oglase, ki so del kampanj. Tako lahko povežemo prikaze oglasa, klike in konverzije znotraj kampanje ter računamo razmerja med prikazi in kliki (CTR) ter kliki in konverzijami (CVR).



Slika 2.4: Diagram oglaševalske strukture.

2.4 Problem optimizacije

V tem diplomskem delu se posvečamo optimizaciji kupovanja oglasnega prostora tako, da izboljšamo uspešnost porabe denarja glede na učinkovito ceno na akvizicijo (CPA). Cilj je znižati CPA in dvigniti CVR na prikaz oglasa. Izkaže se, da je v praksi problem zelo težak. RTB se zelo hitro spreminja, kar pomeni, da morajo biti rešitve prilagodljive glede na trenutno stanje trga. Podatki niso popolnoma zanesljivi, saj je sledenje narejeno preko zahtev pik-

slov iz oglaševalčevih strani, ti pa so lahko postavljeni na napačna mesta oziroma se prožijo ob napačnem trenutku. Zato je potrebno vse piksele, ki bodo del učne množice, ročno preveriti, kje so postavljeni in kateri dogodki jih prožijo. Reševanja problema smo se lotili s kombinacijo seznama dovoljenih kombinacij oglasne borze, spletne strani in identifikatorja oglasnega mesta (XPT) ter seznama prepovedanih uporabniških agentov (angl. *useragent*). Opisu dela in metod se bomo posvetili v poglavju 4.

Poglavje 3

Orodja, metode in podatki

V tem poglavju predstavimo orodja s kratkim opisom, razlago načina uporabe in zakaj smo to orodje izbrali. Opišemo še metode in prijeme, ki smo jih uporabljali v sklopu diplomskega dela, ter strukturo in obdelavo podatkov.

3.1 Orodja

V tem delu se bomo posvetili opisu in razlagi orodij, ki ga uporabljamo. Večina orodij je iz področja podatkovne znanosti. Uporabljena orodja so trenutni standard za obdelavo velikih količin podatkov. Opisana sta *Google BigQuery* in *Python* ter pripadajoče knjižnice, ki so uporabljene za obdelavo podatkov in razvoj programa.

3.1.1 Google BigQuery

BigQuery [3] je *Google*-ovo visoko prilagodljivo podatkovno skladišče za podjetja. *BigQuery* je postavljen na modelu brez strežnikov, kar pomeni, da nima infrastrukture, za katero bi bilo potrebno skrbeti podjetju, ki uporablja to storitev. Omogoča podatkovno analizo na osnovi *SQL*.

V *BigQuery* so shranjeni razčlenjeni dnevnik zahtev, ponudb, obvestil o zmagah, klikih in konverzijah. Vsaka od teh ima svojo tabelo. Preko *SQL* lahko združujemo tabele in sledimo dogajanju od zahteve za prikaz oglasa do

konverzije, če do te pride. Na tabeli, ki jo pridobimo z združevanjem zahtev, odgovorov, obvestil, klikov in konverzij, lahko uporabljamo razne funkcije združevanja, kot so seštevanje in preštevanje. Preko teh lahko dobimo vpogled v podatke, ki nam razkrijejo vzorce ali pa pripravimo množico podatkov za gradnjo modela za strojno učenje.

Za uporabo tega orodja smo se odločili, ker omogoča optimizirano obdelavo velikih količin podatkov po nizki ceni. Ostale prednosti *BigQuery*-ja so, da omogoča varno deljenje vpogledov v podatke med člani organizacije, shranjevanje rezultatov neposredno v *Google Sheets* in gradnjo modelov za strojno učenje.

3.1.2 Python 3

Python [7] je interpretiran, visoko-nivojski, splošno namenski programski jezik. V času pisanja diplomskega seminarja je *Python* najbolj razširjen programski jezik uporabljen v podatkovni znanosti. *Python*-ov dobro razviti sklad za podatkovno znanost ponuja velik razpon knjižnic, ki pomaga pri manipulaciji s podatki, optimizirano analizo, risanjem in prikazovanjem grafov in podobno. Ima tudi razširjeno skupnost uporabnikov, kar poenostavi iskanje rešitev za probleme, ki nastajajo ob programiranju. Kombinacija *Python*-ove filozofije, ki poudarja berljivost kode, in *Jupyter Notebooks*, ki omogočajo delno interpretacijo celic s kodo in podpirajo pisanje kode v prozi podobni obliki, se izkaže za idealno okolje za razvoj programske opreme za obdelavo podatkov, razvijanje teorij in ustvarjanje, učenje ter testiranje modelov za strojno učenje.

Jupyter Notebook [11] je odprto-kodna strežnik-klient spletna aplikacija, ki omogoča ustvarjanje in deljenje dokumentov v obliki beležnice, ki vsebujejo živo kodo, enačbe, vizualizacije in pripovedno besedilo. Aplikacija je odprta v spletnem brskalniku, ki je povezan na strežnik, ki je postavljen na lokalnem računalniku ali pa na oddaljenem strežniku. *Jupyter Notebook* je orodje, ki je primarno namenjeno interaktivnemu razvoju in predstavitvi projektov iz področja podatkovne znanosti. Beležnica *Notebook* združuje kodo

in njene rezultate v en dokument, s čimer omogoči hitro iteracijo v razvoju idej in programa ter omogoča preglednost procesa. Beležnice so sestavljene iz celic, ki vsebujejo programsko kodo. Vsako od teh celic se lahko interpretira posebej, kar nam dovoli, da podatke uvozimo le enkrat, kar skrajša dolžino razvoja, ko so množice podatkov velike, uvožene podatke pa lahko večkrat zaporedno obdelamo na različne načine.

NumPy [4] je knjižnica za *Python*, ki doda podporo za večdimenzionalne matrike *ndarray*, ki temeljijo na programskem jeziku *C*, in velik nabor visokonivojskih matematičnih operacij, ki delujejo na matrikah. Podatkovna struktura *ndarray* se razlikuje od *Python*-skih seznamov tako, da imajo definirano dolžino in elementi znotraj *ndarray* morajo biti vsi istega podatkovnega tipa, ki so prav tako definirani v knjižnici *NumPy*, kot *C*-jevski nabori. Zato nam *NumPy* omogoča optimizirano manipulacijo s podatki. Glavni razlog za uporabo pa je dejstvo, da na *NumPy* temelji knjižnica *pandas*. Zato *NumPy* uporabljamo skupaj z vrsticami *pandas*-ovih *DataFrame*-ov imenovanih *Series*, ki so razširljivi *NumPy ndarray*-i.

Matplotlib [10] je *Python* knjižnica za načrtovanje 2D grafov, ki proizvede visoko-kakovostne slike za publikacije ali interaktivna okolja. Za enostavno risanje grafov ima *Matplotlib* modul *pyplot*, ki z lupino *IPython*[6] zagotavlja *MATLAB*-podoben[16] vmesnik. Ima velik nabor različnih grafov, s katerimi lahko prikažemo podatke in tako lažje opazimo vzorce v podatkih, razne porazdelitve in podobno. Omogoča popolno kontrolo nad stili črt, lastnostmi pisav, lastnostmi osi in podobno preko objektno orientiranega vmesnika ali nabora funkcij. *Matplotlib* lahko uporabimo v *Python* skriptah, *Python* in *IPython* lupinah, *Jupyter Notebook*-ih, strežnikih spletnih aplikacij in grafičnih uporabniških vmesnikih. V našem primeru *Matplotlib* uporabljamo za izris grafov na podlagi danih podatkov. Lahko ga uporabimo na lokalnem računalniku ali pa na strežniku, če količina podatkov presega spomin računalnika.

Pandas [15] je knjižnica za *Python*, ki zagotavlja hitre, prilagodljive in izrazne podatkovne strukture, ki so bile narejene za delo z relacijskimi ali

označenimi podatki na lahek in intuitiven način. Knjižnica je namenjena analizi praktičnih podatkov iz realnega sveta v *Python*-u, kot visoko-nivojski gradnik programa. Podatki, ki so v obliki tabel s heterogenimi vrsticami, kot so v SQL ali v *Excel*-ovih preglednicah, so najbolj primerna za podatkovne strukture, ki jih podpira *pandas*. *Pandas* je tudi primeren za podatke časovnih vrst, ki so lahko urejeni ali neurejeni, in poljubne matrike z označenimi vrsticami in stolpci, ki vsebujejo homogene ali heterogene podatke.

Primarni podatkovni strukturi, ki jih podpira *pandas*, sta *Series* (enodimenzionalen nabor, ki je narejen na osnovi *NumPy ndarray*-ev) in *DataFrame* (dvodimenzionalna matrika, ki je sestavljena iz več *pandas Series*). *DataFrame* ponuja vse funkcije *R*-jevega [8] *data.frame* in še nekaj dodanih funkcij. Kot je bilo omenjeno zgoraj, je *pandas* zgrajen na podlagi *NumPy*, kar zagotavlja hitrejše izvajanje in lažje kodiranje kot čisti *Python*, in s tem namenjen lažji integraciji z drugimi knjižnicami, ki so del znanstveno-računskega okolja *SciPy* [5].

Razlogi za uporabo *pandas* so lahko reševanje manjkajočih podatkov (predstavljenih z *NaN*) v vseh podatkovnih tipih, ki jih definira *pandas*, možnost spreminjanja velikosti podatkovnih struktur, kar nam omogoča brisanje in dodajanje vrstic in stolpcev v *DataFrame* in samodejno ter eksplicitno usklajevanje podatkov. *Pandas* vsebuje zmogljivo in prilagodljivo funkcionalnost za združevanje s *split-apply-combine* operacijami na množicah podatkov za agregacijo in transformacijo podatkov. Omogoča pretvorbo podatkovnih struktur iz *NumPy* in *Python* neposredno v *pandas DataFrame*. Omogoča inteligentno rezanje podatkovnih struktur na podlagi etiket *label-based*, indeksiranje in operacije na podnaborih podatkov, ki jih lahko združujemo na intuitiven način. *Pandas* vsebuje tudi robustna *IO* orodja za uvoz in izvoz podatkov v obliki *CSV*, *XLSX* ali pa neposredno iz *SQL* baze, kar v našem primeru omogoča, da izvožene podatke iz *BigQuery* uvozimo neposredno v *pandas DataFrame* in takoj začnemo z obdelavo.

3.2 Metode

V tem delu se bomo posvetili opisu in razlagi metod in prijemov, ki smo se jih poslužili med raziskovalnim delom. Posvetimo se opisu metod testiranja, iskanja in klasificiranja sledilnih pikslov ter analize podatkov.

3.2.1 Testiranje

Razvoj programske opreme je testno usmerjen. Posledično je sprva več dela, vendar pa se tako izognemo delu z iskanjem in odpravljanjem napak v prihodnosti. Uporabljamo enotske teste in A/B teste v produkciji.

Z enotskimi testi preverjamo manjše dele programa, kot so funkcije, razredi in njihove metode. Namenjeni so temu, da se prepričamo, da pravilno deluje vsaka podenota programa posebej. Ker v splošnem ni vsaka metoda ali razred odvisen samo od sebe, se zgodi, da se lahko test ene enote prelije tudi v drugo, kar lahko služi kot varnostna mreža za produkcijsko implementacijo, saj lahko sprememba v kodi, ki je lokalno pravilna, povzroči probleme v nadaljnjem delu programa. Takim problemom se lahko do neke mere izognemo z enotskimi testi, ki so podobni integracijskim testom, ker preverjajo, kako se ujema več enot med seboj. Enotski testi so večinoma samodejni, kar pospeši razvoj in sporoči napake, ko prevajamo kodo, lahko pa se jih požene tudi ročno.

A/B testi se prvič pojavijo sredi 19. stoletja v farmaciji, kjer so testirali učinkovitost zdravil na ljudeh. V teh testih se je populacija razdelila na dve skupini, kjer je prva bila kontrolna skupina in je prejela placebo, kar pomeni, da se testira, kako se spreminja stanje pacienta brez zdravil, druga pa je prejela testno zdravilo in je bila ob koncu eksperimenta primerjana s prvo skupino. A/B testi so dvojno slepi, kar pomeni, da tako zdravniki kot pacienti ne vedo ali dobivajo placebo ali testno zdravilo. Da je zdravilo učinkovito mora biti razlika v rezultatih statistično značilna. A/B testi pa se dobro obnesejo tudi pri testiranju sprememb pri programiranju na spletnih platformah in storitvah. Za primer lahko vzamemo spremembo grafičnega

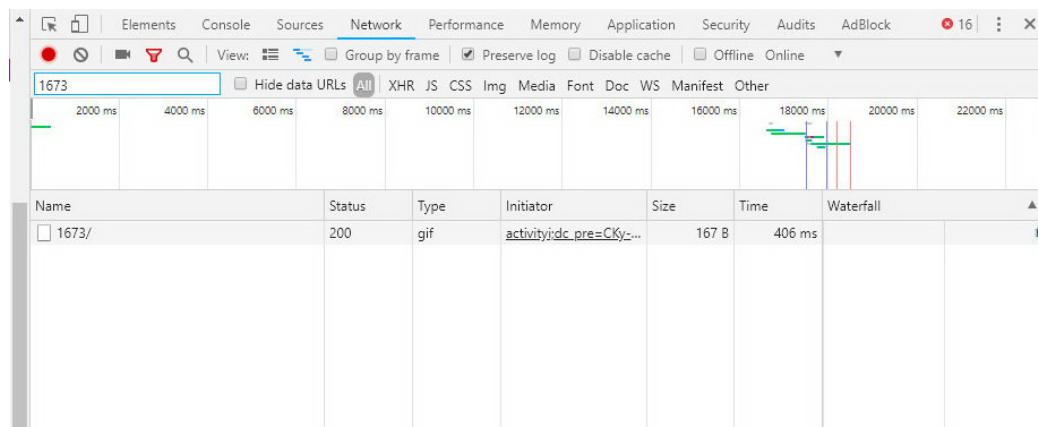
vmesnika za uporabnike, ko prenavljamo spletno stran, ker želimo, da bi uporabniki odprli več strani, ostali dalj časa na naši strani in podobno. To lahko testiramo z A/B testom. Vzporedno postavimo obe obliki spletne strani in naključno razporejamo uporabnike na staro in novo verzijo. Za vsakega uporabnika si beležimo metrike, ki jih želimo optimizirati. Da se izognemo šumu v meritvah, A/B test pustimo v produkciji dalj časa. Po zaključenem testu primerjamo rezultate in se odločimo, katero verzijo strani bomo uporabljali od te točke naprej. Podobno lahko testiramo tudi razne algoritme, kot je na primer algoritem, ki podaja predloge za filme, ki bi lahko zanimali obiskovalca strani.

3.2.2 Iskanje in klasificiranje sledilnih pikslov

Za gradnjo učnega seta podatkov potrebujemo piksle, ki jih je potrebno poiskati na spletnih straneh oglaševalcev in klasificirati. To počnemo, ker ne vemo, katere vrste je piksel, saj tega podatka od oglaševalcev ne dobimo. Izkazuje se tudi, da oglaševalci piksle postavijo na napačna mesta, na primer piksel, ki naj bi bil vezan na nakup, se sproži ob obrazcu za vnos kreditne kartice namesto na zahvalni strani, ki se odpre po zaključenem nakupu.

Najprej iz *BigQuery* pridobimo tabelo, ki povezuje vse od zahteve za dražbo do aktiviranega piksla, za vsako tabelo uporabimo levo pridružitev, da lahko izračunamo razmerje med kliki in konverzijami (CVR). Dobljeno tabelo pretvorimo z združevalnimi funkcijami tako, da dobimo vse piksle, vezane na konverzije, ceno na aktivacijo in CVR posameznega piksla. Seznam pikslov razvstimo po CVR piksla naraščajoče, ker nas zanimajo predvsem težke konverzije, ki imajo nižji CVR. Za vsak piksel vemo, na kateri spletni strani se je sprožil, zato obiščemo to stran in pogledamo, kaj je potrebno narediti, da sprožimo piksel. To lahko vključuje odpiranje podstrani, izpolnjevanje obrazcev, izpolnjevanje *captcha*, klikanje gumbov prisotnih na strani in podobno. Da se je piksel sprožil opazimo tako, da odpremo razvijalska orodja v brskalniku, nato odpremo zavihek za povezavo *Network* (glej sliko 3.1), označimo, da želimo ohraniti dnevnik zahtev, in v iskalno vrstico

vpišemo identifikator piksla. Ko se piksel pojavi v dnevniku povezav, ga lahko klasificiramo glede na to, kaj je bilo potrebno narediti, da se je piksel sprožil.



Slika 3.1: Razvijalska orodja v Google Chrome.

3.2.3 Analiza podatkov

Za analizo podatkov uporabljamo *BigQuery*, ker omogoča hitre poizvedbe po velikih količinah podatkov, in *Python* s knjižnicami. Najprej pripravimo primarno množico podatkov v *BigQuery*, kjer povežemo zahteve za dražbe, odgovore na dražbe, obvestila o zmagah, klike in piksele. Da lahko podatke naložimo v spomin, začnemo s časovno manj obsežnimi seti podatkov. Odpremo interaktivno *Python* sejo z *Jupyter Notebook*-om, naložimo podatke v *pandas DataFrame* in jih začnemo obdelovati. Na tak način lahko najhitreje iteriramo skozi agregacije podatkov, razvijamo teze in iščemo vzorce. Ko se nam prikažejo vzorci, prepisemo zaporedje združevalnih funkcij v *SQL*. Pripravljen *SQL* poženemo v *BigQuery*-ju, na večji množici podatkov in dobimo novo agregirano podatkovno množico. To nato odpremo v *Google Sheets*, narišemo primerne grafe in preverimo, ali opazimo enake vzorce, kot smo jih opazili na manjši množici podatkov.

3.3 Podatki

Vsi dogodki, med katere sodijo zahteve za dražbo, odgovori na zahteve, kliki in piksli za konverzije, se beležijo v dnevniku dogodkov. Dogodki iz dnevnika so periodično obdelani in shranjeni v relacijske tabele na *Google BigQuery*. Vsak tip dogodka ima svoje tabele, saj ima vsak dogodek različno strukturo stolpcev v tabeli. Vsak tip tabele pa je deljen na podtabele, kjer so dogodki razčlenjeni po dnevih. Delitev po dnevih je optimizacija, ki nam omogoča znižati stroške uporabe *BigQuery*, ki zahteva plačilo glede na količino obdelanih podatov. Za našo raziskavo nismo potrebovali celotnih tabel. Pri zahtevah za dražbo nas zanimajo ime borze, ki je poslala zahtevo, naslov založnikove strani in identifikator oglasnega mesta. Odgovore na zahteve za dražbo uporabimo kot povezovalno tabelo med vsemi dogodki. Iz obvestil o zmagi vzamemo ceno prikaza. Iz klikov in pikslov za konverzije pa vzamemo identifikator, ker nas zanima, če je prikaz oglasa privedel do klika ali konverzije.

Poglavje 4

Opis dela

V tem delu se posvetimo opisu raziskovalnega dela diplomske naloge. Predstavljena sta dva pristopa k problemu, ki jo na koncu združimo. Pri opisu dela na seznamu dovoljenih kombinacij oglasne borze, spletne strani in identifikatorja oglasnega mesta (XPT) se osredotočimo na postopek izpeljave smiselne formule za ocenjevanje učinkovitosti kombinacij XPT. Nato pa sledi opis dela na seznamu uporabniških agentov.

4.1 Seznam dovoljenih XPT

Med analizo podatkov se je razvila teza, da nekatera oglasna mesta dajejo v povprečju boljše rezultate za enako ceno. Po izračunu razmerja med kliki in konverzijami (CVR) za vsako kombinacijo XPT je bila teza potrjena, saj so imele nekatere kombinacije višje CVR kot druge, medtem ko na nekaterih kombinacijah nismo zabeležili nobene konverzije. Pristopov za izkoristek te opazke je več. Odločili smo se za gradnjo seznama dovoljenih kombinacij XPT, na katerem bomo kupovali bolj kvaliteten promet. Za gradnjo takega seznama je potrebno definirati, katere kombinacije so dobre in katere so slabe, oziroma uvesti sistem ocenjevanja kombinacij. Pristopov k taki oceni je več. Da bi odkrili najprimernejšo oceno, se poslužujemo iterativnega procesa, kjer začnemo z enostavno oceno, ki jo imenujemo najosnovnejši sprejemljivi

produkt (angl. *MVP*), ki jo postopoma izboljšujemo ali spreminjamo. Cilj raziskovalnega dela je najti najustreznejšo razvrstitev XPT tako, da bomo izboljšali učinkovitost sistema za kampanje s težkimi konverzijami.

4.1.1 Iskanje najustreznejše razvrstitve

4.1.1.1 Razvrščanje po številu konverzij

Za MVP vzamemo razvrščanje XPT kombinacij po številu konverzij, ki smo jih zabeležili na posamezni kombinaciji. Za izračun uporabimo enačbo:

$$ocena_{xpt} = \sum_{c \in C} k_{xpt,c} \quad (4.1)$$

kjer s C označimo množico kampanj in s k označimo zabeležene konverzije. Izkaže se, da to ni dober pristop. Število konverzij nam ničesar ne pove o kvaliteti prometa, ki ga kupujemo na določeni kombinaciji XPT. Na kombinaciji A dobimo več konverzij kot na kombinaciji B, vendar pri kombinaciji A kupimo več prikazov ali pa so ti dražji kot prikazi na kombinaciji B, kar pomeni, da ima kombinacija A višjo cena za akvizicijo (CPA) od kombinacije B in je višje na našem seznamu, kar ni v skladu z našimi cilji. Primer je prikazan v tabeli 4.1.

	Kombinacija A	Kombinacija B
poraba	\$100	\$72
prikazi	1000000	480000
konverizje	20	18
CPA	\$5	\$4

Tabela 4.1: Primer, ki pokaže slabost razvrščanja po številu konverzij.

4.1.1.2 Razvrščanje po CVR

Boljši pristop od razvrščanja po številu konverzij je razvrščanje glede na razmerje klikov in konverzij (CVR). Razmerje izračunamo z enačbo:

$$ocena_{xpt} = \frac{\sum_{c \in C} k_{xpt,c}}{\sum_{c \in C} l_{xpt,c}} \quad (4.2)$$

kjer z l označimo klike, dočim za ostale oznake velja enako kot zgoraj. Vendar ima tudi ta pomanjklivosti. Pri tej razvrstitvi nas zanima, koliko klikov potrebujemo, da pridemo do konverzije, kar pa ni nujno, da se bo odražalo tudi v boljšem CPA. Na primer kombinacija A ima za nekaj odstotkov višji CVR od kombinacije B, a hkrati je CPA na kombinaciji A dvakrat višji od kombinacije B. Iz tega sledi, da ima kombinacija B boljši CPA in je bolj primerna za našo optimizacijo, čeprav je kombinacija A višje na seznamu. Primer prikazan v tabeli 4.2.

	Kombinacija A	Kombinacija B
poraba	\$10000	\$2500
kliki	10000	10000
konverzije	20	10
CPC	\$1	\$0,25
CPA	\$500	\$250
CVR	0,2%	0,1%

Tabela 4.2: Primer, ki pokaže slabost razvrščanja po CVR.

4.1.1.3 Ocena na podlagi primerjave pričakovanih in dobljenih konverzij

Zaradi zgoraj navedenih problemov se odločimo za nov pristop z oceno. Ocena temelji na razmerju med številom konverzij, ki smo jih zabeležili na kombinaciji XPT in številu konverzij, ki smo jih pričakovali, glede na porabljena sredstva, kar izračunamo z enačbo:

$$ocena_{xpt} = \frac{\sum_{c \in C} k_{xpt,c}}{\sum_{c \in C} y_{xpt,c}} \quad (4.3)$$

kjer z y označimo pričakovane konverzije, ostale oznake pa prenesemo iz zgornjih enačb. Vpeljemo formulo za računanje števila pričakovanih konverzij:

$$y_{xpt,c} = \frac{s_{xpt,c} * k_c}{s_c} \quad (4.4)$$

kjer s s označimo porabo denarja, in z njo razširimo enačbo (4.3). Tako dobimo enačbo:

$$ocena_{xpt} = \frac{\sum_{c \in C} k_{xpt,c}}{\sum_{c \in C} \frac{s_{xpt,c} * k_c}{s_c}} \quad (4.5)$$

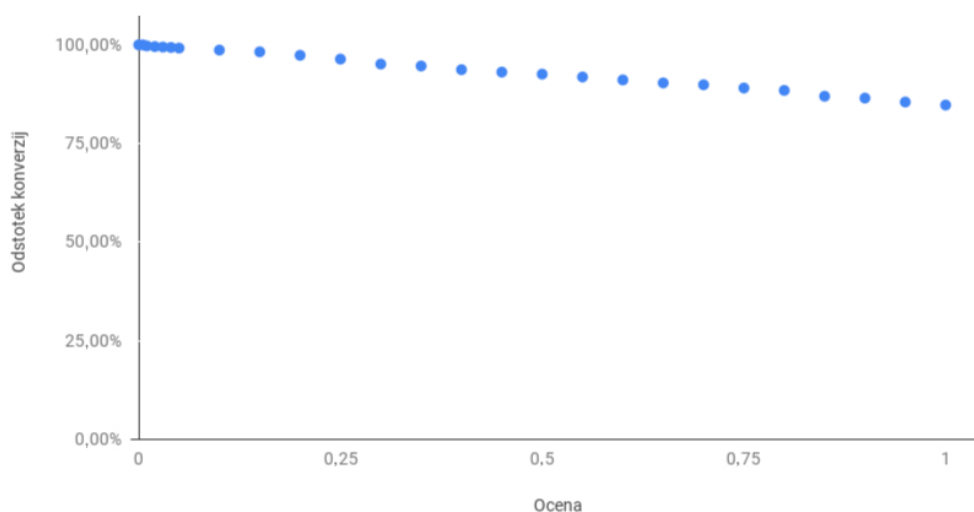
Ta ocena je bolj usmerjena proti optimizaciji za CPA, vendar privzame, da so vse konverzije enakovredne. Na primer vpis naslova za spletno pošto, ki je poceni in se dogaja pogosto, primerja z dragim in redkim dogodkom, kot je nakup avtomobila. Ker s to oceno ne upoštevamo, da so konverzije različne, ne upoštevamo dejstva, da je porazdelitev oglasov, ki jih prikazujemo na različnih kombinacijah XPT, različna. Kar pomeni, da denarna enota porabljenega za nakup oglasov s cenejšo konverzijo, ni vredna enako kot denarna enota porabljenega za nakup dragih konverzij.

4.1.1.4 Ocena na podlagi primerjave pričakovanih in dobljenih konverzij, utežena s porabo

Enačbo (4.5) utežimo s porabo denarja znotraj oglaševalske kampanje na kombinaciji XPT in dobimo enačbo:

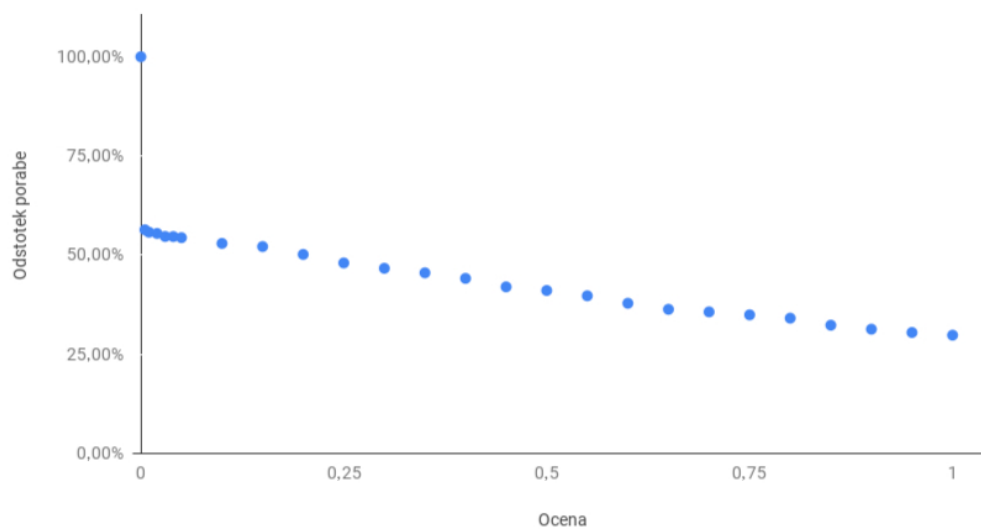
$$ocena_{xpt} = \frac{\sum_{c \in C} k_{xpt,c} * s_{xpt,c}}{\sum_{c \in C} \frac{s_{xpt,c} * k_c}{s_c} * s_{xpt,c}} \quad (4.6)$$

Ta pristop se izkaže za boljšega od neuravnotežene ocene, ker upošteva, da posamezne kampanje nimajo enakega razmerja med porabljenim denarjem in številom konverzij. Vendar pa še vedno primerjamo dve števili, ki nista nujno primerljivi, saj imajo konverzije različno vrednost.

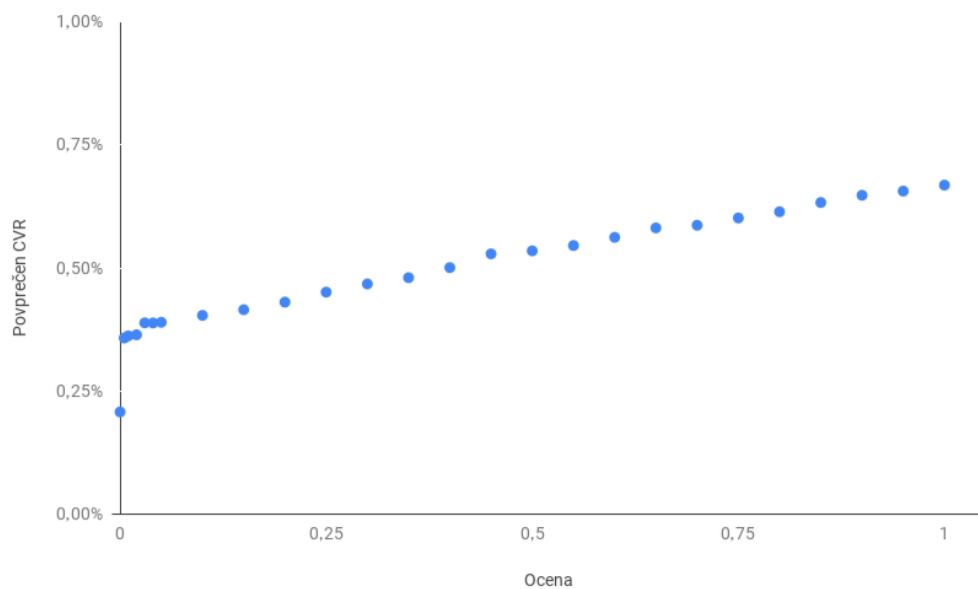


Slika 4.1: Diagram odstotka konverzij glede na višino spodnje meje ocene.

Na diagramu 4.1 lahko opazujemo, kako se spreminja delež konverzij, s spodnjo mejo ocene v primeru, če bi na istem prometu kupovali omejeno s seznamom dovoljenih kombinacij XPT. Število konverzij počasi pada, iz česar lahko zaključimo, da večino konverzij zabeležimo na kvalitetnejših oglasnih mestih. Podobno vidimo na diagramu 4.2, kako se spreminja poraba proračuna. Opazimo, da se poraba skoraj razpolovi, takoj ko uvedemo najnižjo spodnjo mejo. To je posledica tega, da ne moremo slediti uporabniku po kliku na oglas ali pa so oglasna mesta tako slaba, da nikoli niso privedla do konverzij. Na diagramu 4.3 vidimo, kako se spreminja CVR. CVR strmo naraste takoj, ko uvedemo spodnjo mejo, nato pa počasi narašča z višanjem meje. To kaže na to, da višja kot je ocena, kvalitetnejša so mesta za prikaz oglasov.



Slika 4.2: Diagram odstotka porabe glede na višino spodnje meje ocene.



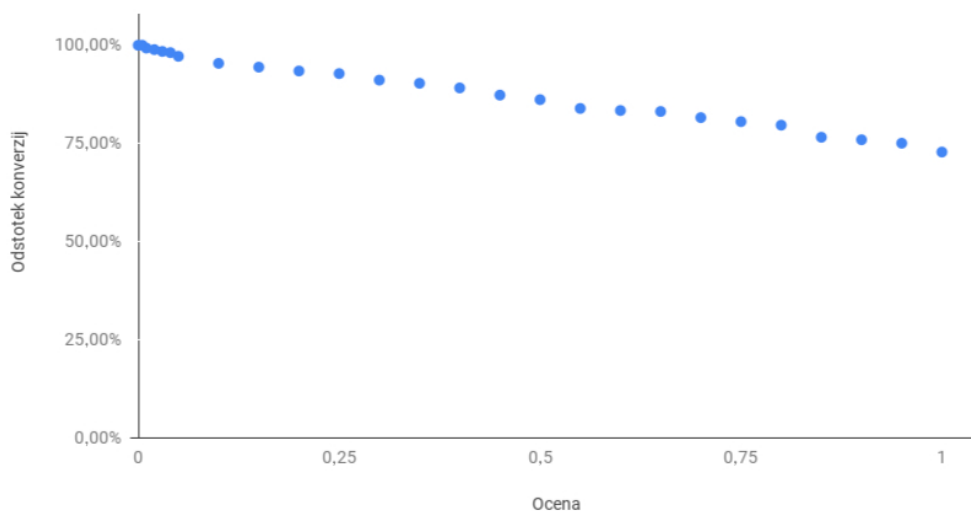
Slika 4.3: Diagram CVR glede na višino spodnje meje ocene.

4.1.1.5 Ocena na podlagi primerjave pričakovanih in dobljenih konverzij, utežena s CPA in porabo

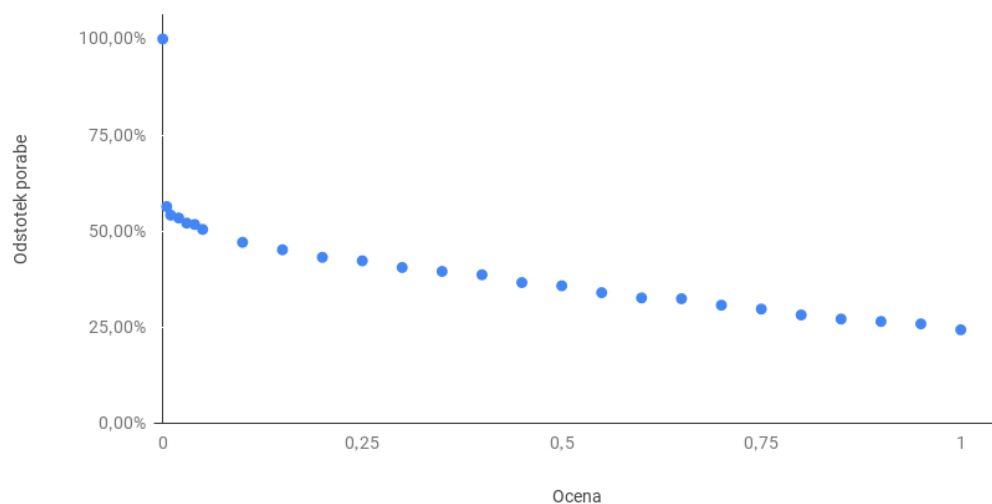
Enačbo (4.6) poskušamo izboljšati tako, da naredimo zabeležene in pričakovane konverzije primerljive. Utežimo enačbo (4.6) uravnotežimo s povprečnim CPA kampanje in dobimo enačbo:

$$ocena_{xpt} = \frac{\sum_{c \in C} k_{xpt,c} * CPA_c * s_{xpt,c}}{\sum_{c \in C} \frac{s_{xpt,c} * k_c}{s_c} * CPA_c * s_{xpt,c}} \quad (4.7)$$

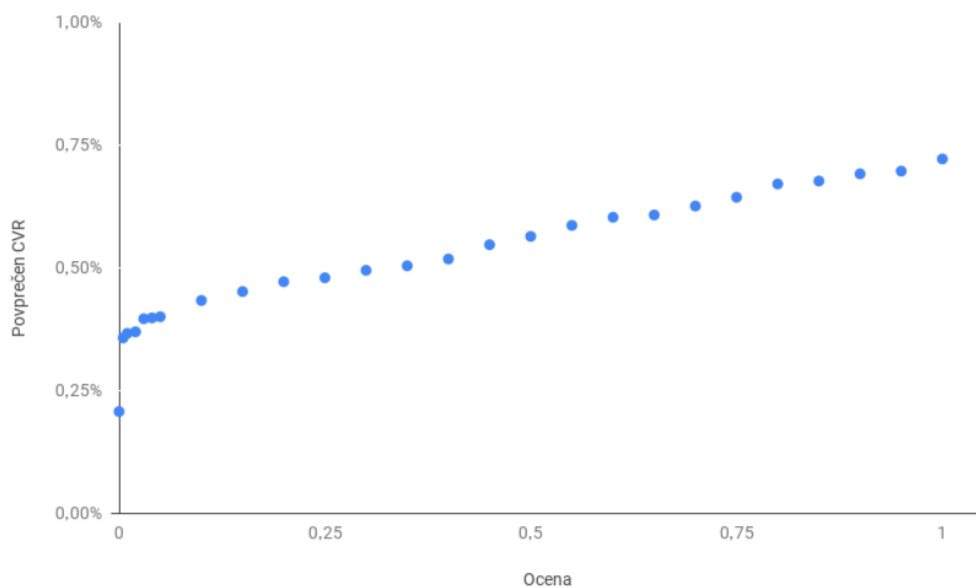
kjer je CPA povprečna cena na akvizicijo v dani kampanji. Tako dobimo dve primerljivi vrednosti. Opazimo, da porabo kampanje na XPT že uporabljamo pri računanju pričakovanega števila konverzij in vrednost zato kvadriramo oziroma upoštevamo dvakrat.



Slika 4.4: Diagram odstotka konverzij glede na višino spodnje meje ocene.



Slika 4.5: Diagram odstotka porabe glede na višino spodnje meje ocene.



Slika 4.6: Diagram CVR glede na višino spodnje meje ocene.

Ko primerjamo diagram 4.1 in diagram 4.4 opazimo, da delež konverzij nekoliko hitreje pada, če oceno računamo z enačbo (4.7) namesto enačbe (4.6). Ko primerjamo diagram 4.2 z diagramom 4.5 ter diagram 4.3 z diagramom 4.6, ugotovimo, da je odstotek porabe nižji, vendar pa dobimo višji

CVR, če oceno računamo z enačbo (4.7). Ker smo izboljšali povprečen CVR, se premikamo v pravi smeri. Kljub temu pa je nižja poraba in manj konverzij problem. Sumimo, da je razlog zato, ker kvadriramo porabo kampanje na posamezni kombinaciji XPT.

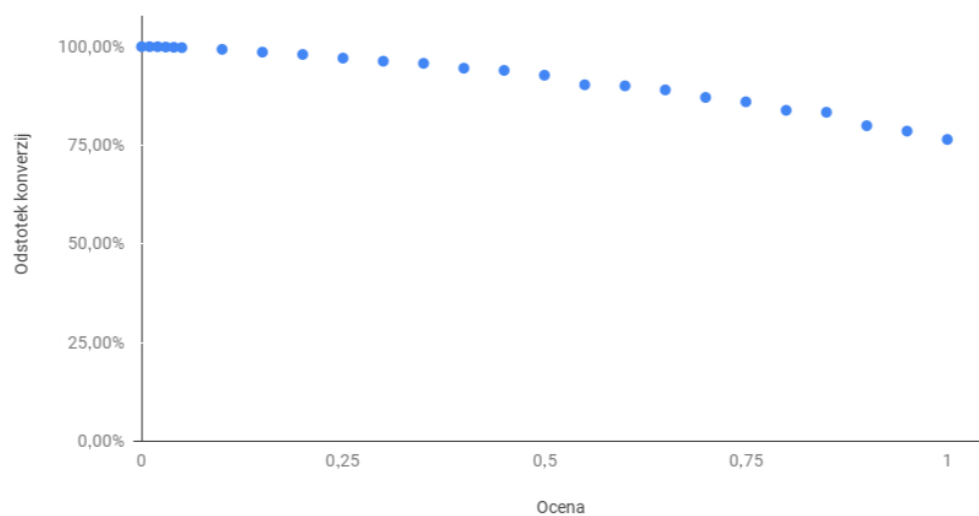
4.1.1.6 Ocena na podlagi primerjave pričakovanih in dobljenih konverzij, utežena s CPA

Iz enačbe (4.7) odstranimo utež s porabo, tako se izognemo kvadriranju, vendar obdržimo primerljivost med števili zabeleženih in pričakovanih konverzij. Dobimo enačbo:

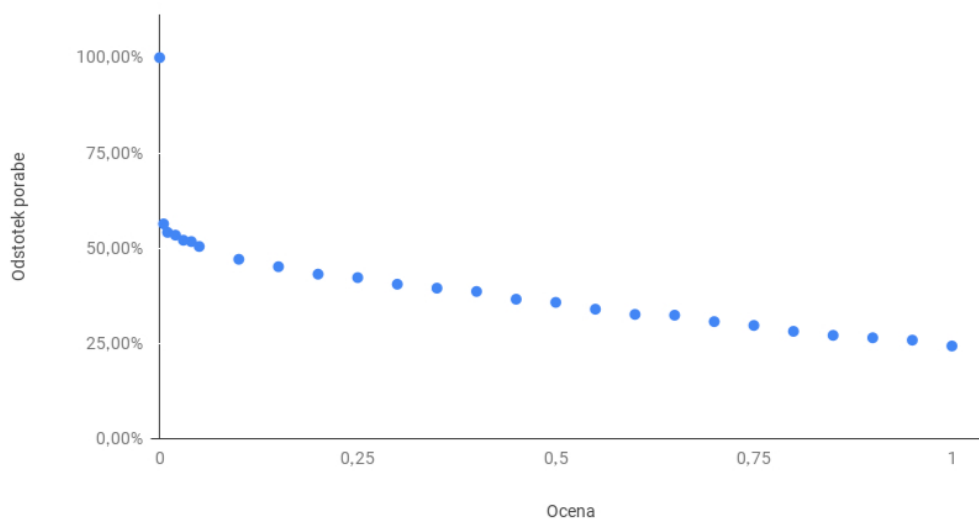
$$ocena_{xpt} = \frac{\sum_{c \in C} k_{xpt,c} * CPA_c}{\sum_{c \in C} \frac{s_{xpt,c} * k_c}{s_c} * CPA_c} \quad (4.8)$$

Opazimo, da z uporabo enačbe (4.8) nekoliko popravimo problem, a ne toliko, kot smo pričakovali. Delež konverzij začne hitreje upadati, ko mejo postavimo na 0,5 ali več (glej diagram 4.7). Ko primerjamo diagrame od ocene (4.7) in ocene (4.8), opazimo, da z novo oceno delež konverzij počasneje upada in ohranimo tudi malenkost več porabe na račun povprečnega CVR.

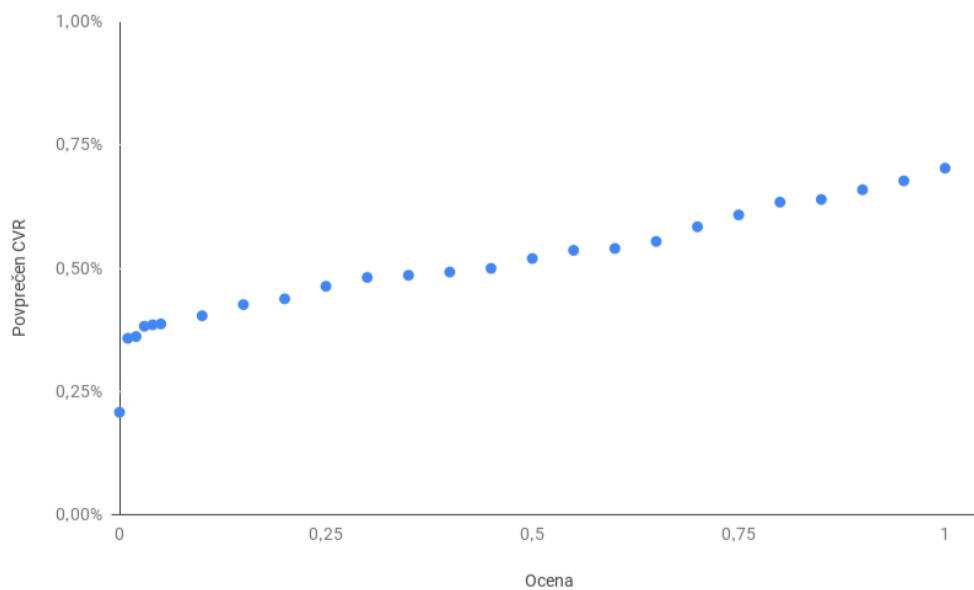
Zato še enkrat temeljito pregledamo podatke in v njih opazimo, da imajo nekatere kampanje previsoke povprečne CPA-je, kot je razvidno na diagramu 4.10. Kampanje na skrajni desni strani narobe utežijo enačbo (4.8), saj množimo s CPA-ji, ki so neprimerno višji od povprečnega CPA. Do tako visokih CPA lahko pride, ker stojijo za drago konverzijo, kot je nakup vozila, ali pa je prišlo do napake na strani oglaševalca, da je narobe postavil piksel, ga odstranil iz strani pred iztekom kampanje in podobno.



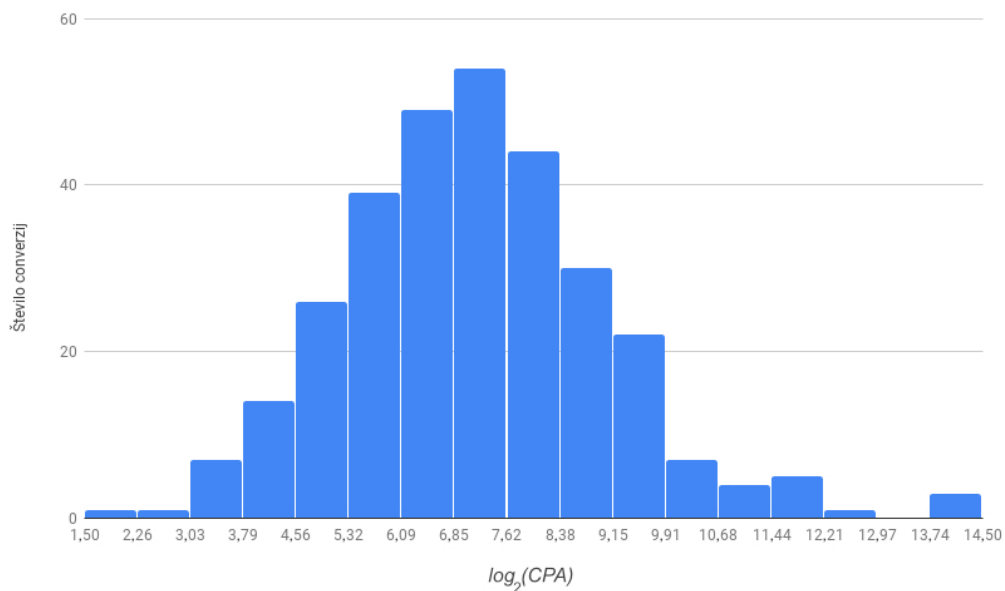
Slika 4.7: Diagram odstotka konverzij glede na višino spodnje meje ocene.



Slika 4.8: Diagram odstotka porabe glede na višino spodnje meje ocene.



Slika 4.9: Diagram CVR glede na višino spodnje meje ocene.



Slika 4.10: Diagram porazdelitve konverzij glede na logaritem CPA z bazo 2.

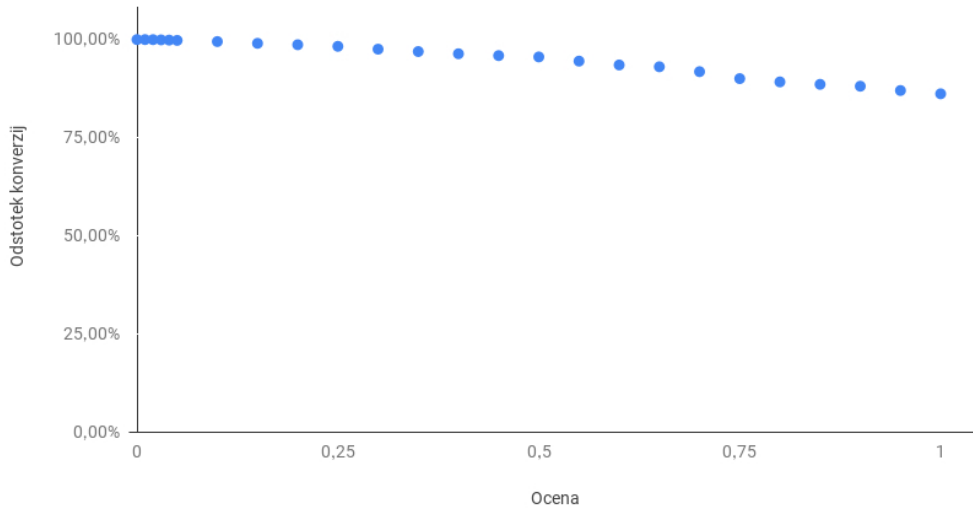
4.1.1.7 Ocena na podlagi primerjave pričakovanih in dobljenih konverzij, utežena z omejenim CPA

Problem previsokih CPA iz enačbe (4.8) rešimo tako, da jih omejimo navzgor s povprečnim CPA, ki ga izračunamo v naši množici podatkov. Vrednosti, ki so nad zgornjo mejo za CPA ponastavimo na povprečno vrednost. Tako dobimo enačbo:

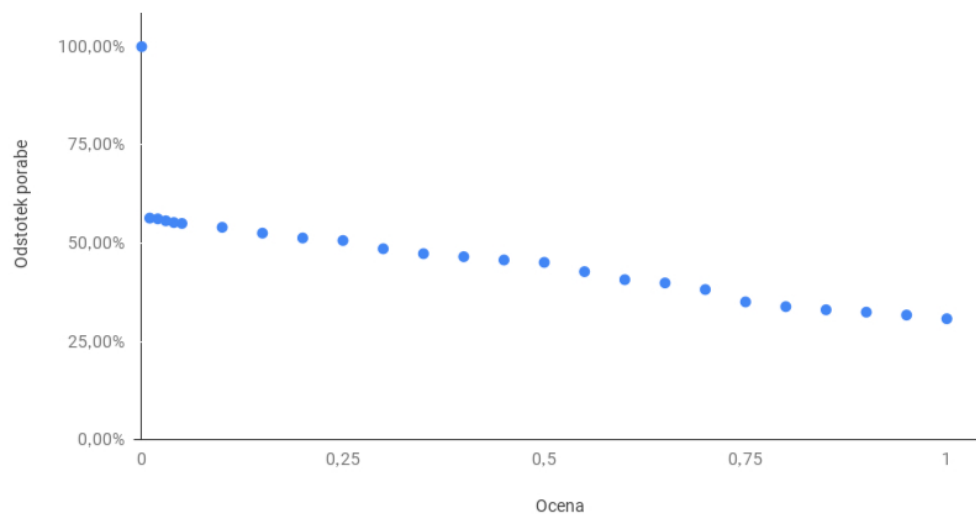
$$ocena_{xpt} = \frac{\sum_{c \in C} k_{xpt,c} * \min(CPA_c, < CPA >)}{\sum_{c \in C} \frac{s_{xpt,c} * k_c}{s_c} * \min(CPA_c, < CPA >)} \quad (4.9)$$

kjer s $< CPA >$ označimo povprečen CPA kampanj. S to oceno se izognemo vsem zgoraj navedenim problemom.

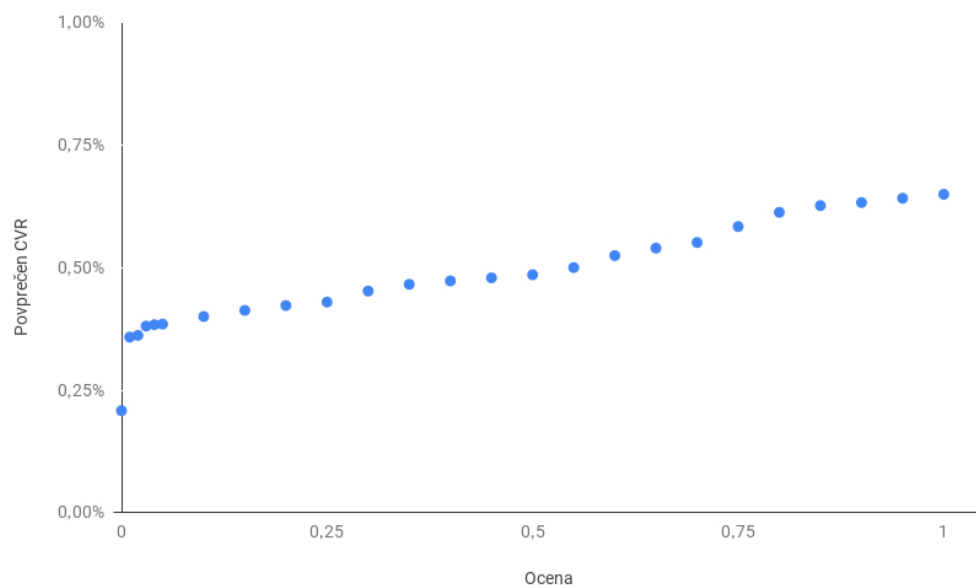
Ob primerjavi diagrama 4.11 z diagramom 4.7 opazimo, da s formulo (4.9) ohranimo več konverzij. Iz primerjave med diagramom 4.12 in diagramom 4.8 opazimo, da za porabo proračunov velja enako. Opazimo tudi, da s formulo (4.9) nekoliko znižamo povprečen CVR v primerjavi s formulo (4.8). Vendar se kljub znižanemu CVR odločimo, da je ocena s formulo (4.9) najbolj smiselna, saj pokrije vse zgoraj naštetе probleme, zato smo tako lahko bolj prepričani v njeno pravilnost.



Slika 4.11: Diagram odstotka konverzij glede na višino spodnje meje ocene.



Slika 4.12: Diagram odstotka porabe glede na višino spodnje meje ocene.



Slika 4.13: Diagram CVR glede na višino spodnje meje ocene.

4.1.2 Določanje spodnje meje

Cilj DSP-jev je učinkovito porabiti dnevne proračune na kampanjah. Uvedba seznama, bi lahko pripeljala do višje učinkovitosti in nižjih cen za konverzije,

vendar bi lahko s tem izgubili delež porabe proračunov. Zato menimo, da ima poskus vpliv tako na uspešnost oglaševalskih kampanj kot tudi na prihodke in dobičkonosnost podjetja. Zato smo se za določitev spodnje meje ocene posvetovali z vodjo produkta, kateremu smo posredovali tabelo 4.3, tabelo 4.4 in grafe 4.11, 4.12 ter 4.13. Na podlagi teh smo se odločili za oceno 0,65, ker ohrani dovolj porabe, da ne škodujemo kampanjam, za katere kupujejo oglasni prostor, hkrati pa jim dvignemo CVR do te mere, da bo poraba njihovih dnevnih proračunov bolj optimalna s stališča CPA. Faktor pri izbiri spodnje meje pa je bilo tudi dejstvo, da za nakupovanje oglasnega prostora uporabljamo algoritem, ki se uči v realnem času, kar pomeni, da je možnost, da porabi celotne dnevne proračune, čeprav ima omejen trg, kjer lahko kupuje, saj tam kupuje več prikazov.

Spodnja meja ocene	Sprememba prikazov	Sprememba klikov	Sprememba konverzij
0,00	0,00%	0,00%	0,00%
0,05	-62,07%	-45,95%	-0,20%
0,10	-62,73%	-48,19%	-0,50%
0,50	-69,10%	-58,95%	-4,41%
0,55	-70,15%	-60,60%	-5,47%
0,60	-72,22%	-62,82%	-6,45%
0,65	-72,74%	-64,06%	-6,91%
0,70	-73,32%	-65,27%	-8,15%
0,75	-75,04%	-67,84%	-9,93%
0,80	-75,84%	-69,64%	-10,76%
0,85	-76,43%	-70,51%	-11,38%
0,90	-76,93%	-70,96%	-11,86%
0,95	-77,89%	-71,71%	-12,95%
1,00	-78,62%	-72,33%	-13,79%

Tabela 4.3: Primerjava deleža prikazov, klikov in konverzij glede na spodnjo mejo ocene v primeru, ko bi kupovali isti promet z istim algoritmom

Tabela 4.3 in tabela 4.4 sta številska prikaza grafov ocene (4.9). V tabeli 4.3 lahko opazimo, da se število konverzij počasi spreminja, medtem ko se ostale metrike spremenijo v večji meri takoj, ko postavimo mejo. Iz tega sledi, da na delu prometa zabeležimo nizko število konverzij, ker jim na določenih oglasnih mestih ne moremo slediti, so nekvalitetna oglasna mesta in ne vodijo do konverzij in podobno. Sistem bi torej izboljšali že samo z uvedbo seznama, ki bi izločil taka oglasna mesta. Kljub temu je naš cilj bolj optimizirati sistem, zato se odločimo za oceno 0,65, ki se izkaže za dober kompromis med izboljšavo CVR in potencialnim znižanjem ostalih faktorjev.

Spodnja meja ocene	Sprememba porabe	Sprememba CPA	CVR
0,00	0,00%	0,00%	0,21%
0,05	-44,92%	-44,81%	0,39%
0,10	-45,91%	-45,63%	0,40%
0,50	-54,81%	-52,73%	0,49%
0,55	-57,17%	-54,69%	0,50%
0,60	-59,20%	-56,39%	0,52%
0,65	-60,04%	-57,07%	0,54%
0,70	-61,71%	-58,31%	0,55%
0,75	-64,85%	-60,98%	0,58%
0,80	-66,05%	-61,96%	0,61%
0,85	-66,86%	-62,60%	0,63%
0,90	-67,46%	-63,08%	0,63%
0,95	-68,19%	-63,46%	0,64%
1,00	-69,13%	-64,19%	0,65%

Tabela 4.4: Primerjava deleža porabe, CPA in CVR glede na spodnjo mejo ocene v primeru, ko bi kupovali isti promet z istim algoritmom

4.1.3 A/B testiranje v produkciji

Ko imamo izbran način ocenjevanja in spodnjo mejo za oceno, lahko sestavimo seznam XPT kombinacij, ki so primerne za CPA optimizirano kupovanje oglasnega prostora. Nato postavimo naslednje hipoteze, izboljšava CVR in CPA, višji CPM in CPC. Ko imamo sestavljen seznam in postavljene hipoteze, lahko v produkcijo implemetiramo A/B test. Zaradi načina delovanja seznama dovoljenih XPT vemo, da bomo izgubili del porabe denarja. Zato se odločimo, da bomo vstopni promet delili tako, da bo testna skupina, ki sodeluje na dražbah za prikazna mesta s seznama, prejela 72% prometa, ostalih 28% prometa pa dobi kontrolna skupina, ki lahko sodeluje na vseh dražbah. Test mora v produkciji ostati daljše časovno obdobje, saj se trg v RTB dnevno spreminja. Na primer razlika med delovnimi in dela prostimi dnevi je očitna. Po tednu dni opazimo, da bili smo pri delitvi prometa preoptimistični, zato ima testna skupina nekaj manj porabe od kontrolne skupine. Zato razmerje spremenimo na 80% za testno skupino in 20% za kontrolno. Seznam smo vsak teden posodobili, da smo ostali v koraku s trgom. V testu je sodeloval le del kampanj, ki so del našega sistema. To so le tiste, v katerih smo našli težke konverzije, in so prejeli odobritev za testiranje od njihovih skrbnikov.

4.2 Seznam prepovedanih uporabniških agentov

Vsak uporabnik interneta ima svojega uporabniškega agenta. Uporabniški agenti niso unikaten identifikatorji posameznikov na internetu. Služijo identifikaciji okolja, v katerem naj bi se naložila spletna stran, ki jo je uporabnik odprl.

Primer uporabniškega agenta:

Mozilla/5.0 (iPad; U; CPU OS 3_2_1 like Mac OS X; en-us) AppleWebKit/531.21.10 (KHTML, like Gecko) Mobile/7B405

Mozilla/5.0 je bil v preteklosti uporabljen kot indikator, da je brskalnik združljiv z mehanizmom za upodabljanje Mozilla. (*iPad; U; CPU OS 3_2_1 like Mac OS X; en-us*) nam pove lastnosti sistema, na katerem je odprt brskalnik. *AppleWebKit/531.21.10* je platforma, ki jo uporablja brskalnik. (*KHTML, like Gecko*) so podrobnosti brskalniskove platforme. *Mobile/7B405* je brskalnikov indikator posebnih izboljšav, ki so na voljo neposredno v brskalniku ali preko tretjih oseb.

Med analizo podatkov smo tudi opazili, da nekateri uporabniški agenti nimajo nobenih konverzij. To se lahko dogaja, ker uporabnikom s takimi agenti ne moremo slediti, recimo, da njihov brskalnik ne dovoli piškotkov, ali pa aplikacija na telefonu odpre ciljno stran oglaševalca v spletnem brskalniku, namesto v aplikaciji sami. Druga možnost pa je, da določen agent pripada robotu za strganje strani (angl. *scraping bot*), ki le klika po oglasih, a ne prispeva nobene konverzije. Zato smo za uporabniške agente naredili seznam prepovedanih agentov. Na seznamu so le tisti agenti, pri katerih nikoli nismo zabeležili nobene konverzije.

Poglavje 5

Rezultati in diskusija

Po treh mescih testiranja smo pridobili rezultate, ki so prikazani v tabeli 5.3. Poraba kontrolne in testne skupine je primerljiva, saj je razlika med skupinama manj kot 5%, kar je bil naš cilj ob vzpostavitvi A/B testa. Prikazov oglasov je bilo občutno manj in to za kar 26,35%. Število klikov zabeleženih klikov pa je bilo bolj podobno, razlika je za 7,67% v prid kontrolne skupine. Iz teh dveh dejstev je očitno, da ima testna skupina občutno višji CTR od kontrole, med tem ko povprečen CPC med skupinama ostane podoben, \$0,19 v testni in \$0,20 v kontrolni skupini. Slednje je pričakovano ob predpostavki, da imamo sistem, ki skozi dan prireja ponudbe tako, da imajo kampanje ob koncu dneva približno enako povprečen CPC, kot si ga želi lastnik kampanje. Razliko pa opazimo tudi med številom konverzij, testna skupina jih je zabeležila 661, kontrolna skupina pa jih je prejela 593, kar je 11,47% izboljšanje na strani testne skupine. Iz tega sledi, da ima testna skupina boljši tako povprečen CPA kot tudi povprečen CVR od kontrolne skupine. Iz rezultatov je razvidno tudi, da testna skupina doprinese 12,65% več konverzij, kot smo jih pričakovali. Kontrolna skupina pa jih zagotovi 4,58% premalo. Pričakovane konverzije so bile izračunane z enačbo (4.4), pri izračunu pa je bil upoštevan celoten promet, zato ima kontrolna skupina negativno vrednost namesto 0. Na podlagi rezultatov iz tabele 5.3 menimo, da je bil poskus uspešen.

Ena od zanimivosti, ki jih razkrivajo rezultati poskusa, je opazka, da

imajo uporabniki, ki vidijo oglas na kombinaciji XPT iz dovoljenega seznama, za 46,18% višji CTR kot uporabniki iz celotnega prometa. Izboljšava CTR se pozna pri povprečnih CPC skupin, ki sta zato skoraj enaka, kljub temu da je povprečen CPM v testni skupini \$0,72 in \$0,51 v kontrolni skupini, kar je kar 41,18% razlika. To opazko lahko pojasnimo z dejstvom, da imamo o uporabnikih, ki jih srečamo na bolj prestižnih založniških straneh, več podatkov. Na podlagi teh podatkov pa jim lahko ponujamo boljše oglase oziroma njim bolj zanimive oglase, za katere je višja verjetnost, da bodo privedli do klika.

Zajete podatke iz produkcijskega A/B testa razdelimo po kampanjah in primerjamo CVR med testno in kontrolno skupino, kot je prikazano v tabeli 5.1 in tabeli 5.2. Na teh podatkih naredimo sledeč t -test:

$H_0: <CVR>_{kontrola} = <CVR>_{test}$ (povprečna CVR vzorcev sta enaka)

$H_1: <CVR>_{kontrola} \neq <CVR>_{test}$ (povprečna CVR vzorcev nista enaka)

Testna in kontrolna skupina sta v tem primeru odvisna ena od druge, saj črpata denar iz istega dnevnega proračuna, kar pomeni, da kar porabi prva, druga ne mora in obratno. Zato se odločimo za t -test za odvisne vzorce pri $\alpha = 0,05$, v testu pa je 31 kampanj. Izkaže se, da je $t = 3,1841$ kar je več od kritične vrednosti 2,042. Ničelno hipotezo zavrnamo, saj so rezultati testa statistično značilni pri $p = 0,05$.

Kampanja	CVR kontrolne skupine	CVR testne skupine
1	0.11%	0.36%
2	0.95%	0.77%
3	0.00%	2.53%
4	1.37%	1.52%
5	1.38%	1.29%

Tabela 5.1: Tabela CVR kampanj, ki so sodelovale v A/B testu

Kampanja	CVR kontrolne skupine	CVR testne skupine
6	0.20%	0.69%
7	1.78%	2.65%
8	0.50%	1.04%
9	0.48%	2.94%
10	0.98%	2.13%
11	0.93%	2.78%
12	1.99%	8.33%
13	0.65%	0.00%
14	0.20%	3.39%
15	1.30%	3.48%
16	0.57%	1.27%
17	2.06%	0.00%
18	0.82%	0.21%
19	2.77%	4.96%
20	1.23%	0.62%
21	1.52%	2.16%
22	0.00%	1.20%
23	0.66%	1.14%
24	1.18%	4.84%
25	3.95%	1.87%
26	2.20%	8.00%
27	2.50%	1.23%
28	0.00%	1.60%
29	1.04%	2.70%
30	1.31%	2.77%
31	0.75%	0.83%

Tabela 5.2: Tabela CVR kampanj, ki so sodelovale v A/B testu

Opazimo pa tudi morebiten problem. Če pomnožimo porabo denarja obeh skupin tako, da dobimo približek porabe, če bi imela vsaka skupina

	Testna skupina	Kontrolna skupina
poraba	\$45271,27	\$43343,64
prikazi	63081162	85644687
kliki	238104	221144
konverzije	661	593
pričakovane konverzije	586,77	621,48
izboljšava konverzij	12,65%	-4,58%
povprečen CPM	\$0,72	\$0,51
povprečen CPC	\$0,19	\$0,20
povprečen CPA	\$68,49	\$73,09
povprečen CVR	0,28%	0,27%

Tabela 5.3: Rezultati A/B testa (primerjava kontrolne in testne skupine).

celoten proračun oziroma bi le ena od teh skupin bila v produkciji, lahko opazimo, da se poraba zniža za več kot smo pričakovali. Vendar je to le teoretično in s predpostavko, da kupujemo enak promet brez prilagajanja algoritma. Kar pa ne velja za naš sistem, saj se algoritmi za kupovanje na dražbah prilagajajo trgu, kar pomeni, da lahko kompenzirajo tudi za to, da morajo dnevne proračune zapolniti na omejenem delu trga.

Poglavje 6

Zaključek in prihodnje delo

Poudarek naloge je na podatkovni znanosti v RTB. Najprej smo se posvetili raziskavi same domene, da bi poglobili svoje znanje v ozadje RTB-ja. Nato smo se lotili obdelave podatkov z orodji, ki so primerna za tolikšno količino podatkov. Največ časa smo posvetili razvoju smiselne ocene za delitev založnikov na dobre in slabe. Nato pa smo sistem testirali v trimesečnem A/B testu v produkciji na realnih dražbah.

Zaradi pozitivnega vpliva poskusa na CPA in CVR, je bil sistem kupovanja s pomočjo seznamov implementiran kot del produkta. Skrbnikom kampanj smo omogočili, da se odločijo ali bodo v kupovanje kampanje uvedli sistem s seznamami, glede na ključne cilje kampanje. V primeru, ko je cilj kampanje, da ne preseže določenega povprečnega CPA-ja, lahko z vklopom sistema za kupovanje s seznamami izboljšamo uspešnost kampanj in tako strankam zagotovimo višji donos na porabo denarja za oglaševanje. Množica podatkov, ki je bila zbrana med raziskovalnim delom, pa se bo uporabljala kot osnova za razvoj novega optimizacijskega sistema na podlagi KPI-jev, ki jih bodo izbrale stranke. V prihodnje bo potrebno vzpostaviti sistem, ki bo samodejno posodabljal sezname, in rešiti problem raziskovanja potencialnega trga, saj kupovanje s seznamami po definiciji omejuje promet, ki ga lahko kupimo. Zato algoritem ne bi mogel odkriti novih založnikov, ki doprinašajo dovolj vrednosti, da se jih doda na seznam dovoljenih XPT kombinacij. Med

testiranjem tega problema nismo imeli, ker je trg raziskovala kontrolna skupina, testna skupina pa je le žela rezultate te raziskave. Ena od možnih rešitev je, da tudi kampanje, ki naj bi kupovale le s pomočjo seznamov, z nizko verjetnostjo kupujejo tudi izven seznamov in tako raziskujejo.

V primeru, da se množica podatkov, ki smo jo uporabljali za gradnjo seznamov, dovolj razširi, da bi na njej lahko učili modele strojnega učenja, bi lahko razvili model za napovedovanje CVR [2]. Tak algoritem bi lahko bil vzporedno implemetiran z obstoječim algoritmom, ki napoveduje CTR, in bi se ga uporabljalo na kampanjah, na katerih želimo optimizirati proti boljšim CPA. Druga možnost je, da bi zamenjali algoritme v primeru, da bi bila večina kampanj, ki so v našem sistemu, takih, ki želijo nižje CPA [9].

Kot priložnost za optimiziranje sistema pa se izkažejo tudi dražbe prve cene. Čeprav je večina RTB dražb dražb druge cene, se izkaže, da je del prometa tudi dražb prve cene. Sistem bi lahko dodatno optimizirali za dražbe prve cene, saj se ideologiji obeh sistemov razlikujeta. Na dražbah druge cene si lahko privoščimo višjo ponudbo, oziroma lahko ponudimo ponudbo v višini vrednosti prikaza, ki jo izračunamo, saj ponavadi ni potrebno plačati tako visoke ponudbe. Ker na dražbah prve cene plačamo ponudbo v celoti, je možnost, da bi lahko potencialno kupovali promet po boljši ceni, kar bi lahko izboljšalo tako povprečne CPC-je in tudi CPA-je. Zato bi lahko v prihodnje naredili nov algoritem samo za dražbe prve cene, ki bi bil implementiran vzporedno s trenutnim algoritmom. Pristopov za to je več, na primer ocenjevanje razmerja med našo ponudbo in verjetnostjo, da bomo zmagali na dražbi, in na podlagi tega znižamo ponudbo do točke, ko za manj denarja prejmemo enako vrednost [13].

V tej nalogi smo se ob delu na Zemanti srečali z RTB-jem, ki je najhitrejši trg na svetu. RTB je kompleksna domena za podatkovno znanost kljub obsežni količini podatkov, saj se v njej srečamo z mnogimi problemi, ki so vredni raziskave. Denimo reševanje problemov s cenzuriranimi podatki [19], ki nastajajo zaradi tega, ker po eni strani ne dobimo obvestil s strani borz, če smo dražbo izgubili, in po drugi strani, če ima prikaz postavljeno najnižjo

sprejemljivo ponudbo (angl. *bid floor*), tudi lahko ne izvemo, kolikšna je bila druga najvišja ponudba. V tej nalogi smo se posvetili optimiziranju sistema za nižje CPA, iz česar sledi boljša poraba denarja za oglaševanje. Razvoj sistema za cenjevanje kombinacij XPT, ki oplemeniti obsetoječe podatke tako, da lahko s seznamom izluščimo najboljše založnike, je bil najpomembnejši del raziskovalnega dela.

Ker so bili rezultati A/B testa statistično značilni in je imel sam test pozitiven vpliv na delovanje sistema, je bil trenuten sistem razširjen tako, da omogoča skrbnikom kampanj, da omejijo kupovanje oglasnega prostora s seznamami. Tako si lahko na svojih kampanjah zagotovijo višjo vrednost, ki jo dobijo s porabo denarja za oglaševanje. S to razširitvijo smo pozitivno vplivali na učinkovitost sistema za pridobivanje konverzij, zadovoljstvo strank z Zemantininim sistemom in na donosnost podjetja. Torej je največji doprinos diplomskega dela izboljšava industrijskega sistema, ki doprinaša vrednost realnim strankam.

Literatura

- [1] Interactive Advertising Bureau. OpenRTB (Real-Time Bidding). Dosegljivo: <https://www.iab.com/guidelines/real-time-bidding-rtb-project/>, 2019. [Dostopano 7. 1. 2019].
- [2] Kuang chih Lee, Burkay Orten, Ali Dasdan, and Wentong Li. Estimating conversion rate in display advertising from past performance data. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 768–776, New York, NY, USA, 2012. ACM.
- [3] Google Cloud. Google bigquery. Dosegljivo: <https://cloud.google.com/bigquery/>, 2019. [Dostopano: 7. 1. 2019].
- [4] NumPy developers. numpy. Dosegljivo: <http://www.numpy.org/>, 2018. [Dostopano: 7. 1. 2019].
- [5] SciPy developers. Scipy. Dosegljivo: <https://www.scipy.org/>, 2019. [Dostopano: 7. 1. 2019].
- [6] IPython development team. ipython. Dosegljivo: <https://ipython.org/>, 2019. [Dostopano: 7. 1. 2019].
- [7] Python Software Foundation. Python. Dosegljivo: <https://www.python.org/>, 2019. [Dostopano: 7. 1. 2019].
- [8] The R Foundation. R. Dosegljivo: <https://www.r-project.org/>, 2019. [Dostopano: 7. 1. 2019].

-
- [9] Yu (Jeffrey) Hu, Jiwoong Shin, and Zhulei Tang. Incentive problems in performance-based online advertising pricing: Cost per click vs. cost per action. *Management Science*, 62(7):2022–2038, 2016.
 - [10] John Hunter, Darren Dale, Eric Firing, Michael Droettboom, and the Matplotlib development team. matplotlib. Dosegljivo: <https://matplotlib.org/>, 2018. [Dostopano: 7. 1. 2019].
 - [11] Project Jupyter. jupyter. Dosegljivo: <https://jupyter.org/>, 2019. [Dostopano: 7. 1. 2019].
 - [12] Juanjuan Li, Xiaochun Ni, Yong Yuan, Rui Qin, Xiao Wang, and Fei-Yue Wang. The impact of reserve price on publisher revenue in real-time bidding advertising markets. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2017.
 - [13] Xiang Li and Devin Guan. Programmatic buying bidding strategies with win rate and winning price estimation in real time mobile advertising. In Vincent S. Tseng, Tu Bao Ho, Zhi-Hua Zhou, Arbee L. P. Chen, and Hung-Yu Kao, editors, *Advances in Knowledge Discovery and Data Mining*, pages 447–460, Cham, 2014. Springer International Publishing.
 - [14] S. Muthukrishnan. Ad exchanges: Research issues. In Stefano Leonardi, editor, *Internet and Network Economics*, pages 1–12, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
 - [15] pandas. Dosegljivo: <https://pandas.pydata.org/>, 2019. [Dostopano: 7. 1. 2019].
 - [16] Inc. The MathWorks. Matlab. Dosegljivo: <https://www.mathworks.com/products/matlab.html>, 2019. [Dostopano: 7. 1. 2019].
 - [17] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961.

-
- [18] Jun Wang, Weinan Zhang, and Shuai Yuan. Display advertising with real-time bidding (RTB) and behavioural targeting. *CoRR*, abs/1610.03013, 2016.
 - [19] Wush Chi-Hsuan Wu, Mi-Yen Yeh, and Ming-Syan Chen. Predicting winning price in real time bidding with censored data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1305–1314, New York, NY, USA, 2015. ACM.
 - [20] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. Real-time bidding for online advertising: Measurement and analysis. *CoRR*, abs/1306.6542, 2013.
 - [21] Yong Yuan, Feiyue Wang, Juanjuan Li, and Rui Qin. A survey on real time bidding advertising. In *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics*, pages 419–423, Qingdao, China, 2014. IEEE.
 - [22] Weinan Zhang, Shuai Yuan, and Jun Wang. Optimal real-time bidding for display advertising. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1077–1086, New York, NY, USA, 2014. ACM.